Stock Market Forecasting by Integrating Time-Series and Textual Information

Fung Pui Cheong Gabriel

A Dissertation Submitted in Partial Fulfilment

of the Requirements for the Degree of

Master of Philosophy

in

Systems Engineering and Engineering Management

June 2003

Abstract

From the stock market to the commodities market, Hong Kong's financial markets provide the opportunity for everyone to be involved in the local and international economy. In the past decade, scholars from different disciplines, such as psychology, sociology and engineering, have already suggested that events broadcasted by the mass media are highly related to the financial price movements.

This dissertation focuses on investigating the impacts of the mass media on the financial market based solely on time series and text mining techniques. In particular, real-time news articles and intra-day stock prices are used to denote the events broadcasted by the mass media and the stock price movements, respectively. These data are chosen as they are readily available, and the evaluation results obtained can easily be verified.

Several novel data mining and text mining techniques are proposed in this dissertation. The tertiary movements on the stock series are found by a new piecewise linear segmentation algorithm. In addition, a heuristics for selecting news articles that are highly related to price movements is presented. The relationship between news articles and price movements are learned through a novel classification approach, Discriminative Category Matching (DCM). DCM is an efficient and effective classifier which does not need to formulate any sophisticated model but only requires simple statistical data.

Extensive experiments are conducted to evaluate various aspect of the proposed framework using both the benchmark data and real-life data. Encouraging results are obtained which indicate that the proposed framework is highly feasible.

摘要

在現今生活中，每一個人都直接或間接地牽涉到國家金融體系當中。過去，不少來自不同界別的學者，如：心理學，社會學，和工程學等，均指出大眾傳媒的報道與金融市場的價格浮動之間存在一定程度的相互關係。

這篇論文旨在嘗試以數據發掘和文字發掘的技術，來研究大眾傳媒對於金融市場的影響。當中，大眾傳媒和金融市場分別以即時新聞和即時股價來代表。挑選這兩種數據的原因是因爲它們較容易得到，並且以這些數據所得出的實驗結果是較容易驗證的。

在這篇論文中，我提出了很多新穎的數據發掘和文字發掘的技巧。首先，我用了一個全新的直線分段分割法來找出股票的走勢。其次，我亦提出了一個用以篩選那些對於股票走勢有影響的新聞啓發法。另外，我用一個全新的分類法：區別類目相配法，來找出股票的走勢和新聞內容的關係。區別類目相配法不但是一個有效率的分類法，並且能夠精確地把數據分類。它不需要依賴複雜的模型，只需要做一些簡單的統計分析即可運作。

最後，我們用基準數據和眞實數據做了大量的實驗來鑑證這個構思。令人鼓勵的實驗結果指出這個構思是高度可行的。

Acknowledgement

I would like to thank my supervisor, Prof. Jeffrey Yu, for his teaching and encouragements throughout these two years. Without his help, this dissertation could hardly be finished.

Family members have been a special source of encouragement and support. My life and my work have been blessed by them. Thank you their love, patience and good humor while I was pursuing this Master degree.

Special thanks goes to Fiona Choi, who shares life's ups and downs with me. Her beautiful smile is a constant reminder of what life is really all about. Thank you for her proofreading on this dissertation also (of cause, any mistakes remind are the responsibility of the author solely!).

Thanks also go to: Paul Fung for sharing his political opinions with me; Jelly Cheung for her sweet water; Kenny Ho for sharing his Java experiences with me; and Phyllis Leung for her technical supports.

Many others, too numerous to mention, have touched my life throughout these two years. Each played his or her own part in making me the person I am today. They all deserve my thanks for various reasons. Some may be surprised to see their names here, but I have listed them to give my sincere gratitude. They are: Prof. Shu-Cherng Fang, Prof. Wai Lam, Prof Hong-Jun Lu, Prof. Manu Madan and Prof. Helen Meng; Christina Kwok in USA and Echo Leung in Australia; my colleagues: Kaka Huang, Gatien Wong, Su and Angie Lim; Alice Chan, Amy Cheung and Rebecca Chiu.

Fung Pui Cheong Gabriel

The Chinese University of Hong Kong

June 2003

Contents

List of Figures

List of Tables

Part I

The Very Beginning

What registers in the stock market's fluctuations are not the events themselves but the human reactions to these events, how millions of individual men and women feel these happenings may affect the future. Above all else, in other words, the stock market is people.

Bernard Baruch

Chapter 1

Introduction

From the stock market, where securities are bought and sold, to the commodities market, where crops, livestock and metals futures are traded, Hong Kong's financial markets provide the opportunity for almost anyone to become involved in a slice of the local and international economy, either directly or indirectly. Professionals from different disciplines have published quite a lot about the inherent dynamics that drive movements in the prices of financial assets.

In short, the markets' movements are the consequences of the actions taken by investors on how they perceive the events surrounding them and the financial markets. Although human react to the plethora of events and information in a highly subjective manner, their behaviors are usually rational and understandable. In fact, the investors' decisions are greatly influenced by what others are saying and doing within the financial markets. Theories of human behaviors that stress how individual perceive and interpret situations, as well as those underline the dynamics of social influence process, would be helpful in accounting for the human factors that affect the markets' movements.

Nevertheless, one of the most significant impact on our attitudes, beliefs and behaviors comes from the mass media. Scholars from different academic fields, especially from psychology and sociology, have suggested or developed many well-defined theories to account for the influences of mass media. The mass media have a close relationship with our behaviors; meanwhile our decisions on bidding and asking in the financial markets affect price movements. In other words, it is possible to predict the markets' movements by analyzing the impact of the events broadcasted by the mass media.

1.1 Contributions

This dissertation focuses on the problem of predicting the impacts of mass media on financial markets based on data mining techniques. In contrast to the traditional time series analysis, where predictions are made based solely on the historical performance of the time series, here, predictions are made according to non-quantifiable information – textual documents. This requires techniques for mining time series and textual documents concurrently. Nowadays, an increasing number of researches [16, 30, 43, 53, 62] are conducted in this direction. The unique features of the proposed system are summarized below:

- A new piecewise linear approximation algorithm, called t-test based split and merge segmentation algorithm, is applied for finding out the tertiary movements on the financial markets;
- A selection heuristic for selecting positive training examples based on χ^2 estimation on the keywords distribution over the entire document collection and a filtering algorithm for pruning valueless documents is presented;

- A novel, efficient and effective classification approach, called Discriminative Category Matching (DCM), is proposed for learning the relationship between textual documents and markets' movements.

1.2 Dissertation Organization

The content of this dissertation is a further development of the works reported in the following conference papers and book chapter:

1. G. P. C. Fung, J. X. Yu and W. Lam. News sensitive stock trend prediction. In *Proceedings of the 6th International Conference of Pacific-Asia Knowledge Discovery and Data Mining*, pages 481-493, Taipei, Taiwan, 2002.

2. G. P. C. Fung, J. X. Yu and H. Lu. Discriminative category matching: Efficient text classification. In *Proceedings of the 2nd IEEE International Conference on Data Mining*, pages 187-194, Maebashi City, Japan, 2002.

3. G. P. C. Fung, J. X. Yu and W. Lam. Stock prediction: A non-quantifiable approach using real-time news. In *Proceedings of the 7th IEEE International Conference on Computational Intelligence for Financial Engineers*, pages 395-402 Hong Kong, China, 2003.

4. G. P. C. Fung, J. X. Yu and H. Lu. Classify High Speed Text Streams. To be appear in *the 4th International Conference on Web-Age Information Management*, Chengdu, China, 2003.

5. G. P. C. Fung, J. X. Yu and W. Lam. Automatic Stock Trend Prediction by Real Time News. In W. Ching and M. K. Ng, editors, *Advances in Data*

Mining and Modeling, pages 48-59, World Scientific Publishing Co. Pte. Ltd., ISBN: 981-238-354-9, 2003.

This dissertation is divided into five parts. Part I introduces and formulates the problem described in this dissertation. Part II presents the motivations of this research, as well as reviews the techniques presented in the recent literatures that are highly related to this research. Part III proposes a prediction framework, which includes time series representation, text pre-processing, system learning and operation. Part IV evaluates the proposed approach and discusses the experimental results obtained. Finally, Part V summarizes and concludes this dissertation.

Chapter 2

Problem Formulation

A prediction problem involving financial time series and textual documents can be formulated in several ways, such as predicting price level changes based on some newsgroup messages or forecasting future traded volumes according to company profiles. In order to have a concrete idea for what this piece of research is focussed on, this chapter defines the problem statement and outlines a general framework for tackling it.

2.1 Defining the Prediction Task

This dissertation focuses on predicting the impacts of the events broadcasted by mass media on the market's movements. The predictions are made based solely on time series mining and text mining techniques.

For any prediction systems to operate successfully, we first have to archive and label some sets of data and present them to a system for learning their relationships. We call these data as training data. Real-time news articles and

intra-day stock prices are served as training data. They denote the events broadcasted by the mass media and the market's movements, respectively. These data are chosen because they are readily available and the evaluation results obtained can easily be verified.

Three major tasks are defined:

1. Figure out the tertiary movements on the stock prices. A tertiary movement lasts for less than three weeks. Such kind of movement usually denotes the short-term market behavior as well as reflects investors' thought. In other words, this movement is highly affected by the events surrounding the financial market.

2. Detect the relationship between the events mentioned in news articles and the tertiary movements of stock prices. The events detected are served as the major element for training the proposed system.

3. Predict the impact of a newly released news article on the stock price movement. Three kinds of impact are defined: positive, negative and natural. A piece of news article is said to have positive (or negative) effect if the stock price rised (or dropped) significantly for a period after the news article is released. Otherwise, if the news article does not contribute to the stock price fluctuation, the impact of the news article is known as natural.

2.2 Overview of the System Architecture

Figure 2.1 shows the general framework of the system using the Unified Model Language™ (UML)[1].

Figure 2.1 (a) and Figure 2.1 (b) show the training phase and operation phase, respectively. The training phase includes six main procedures:

1. **Figure Out the Tertiary Movements** This process identifies all of the tertiary movements on stock prices. Tertiary movement denotes the short-term market behavior as well as reflects the investors thought.

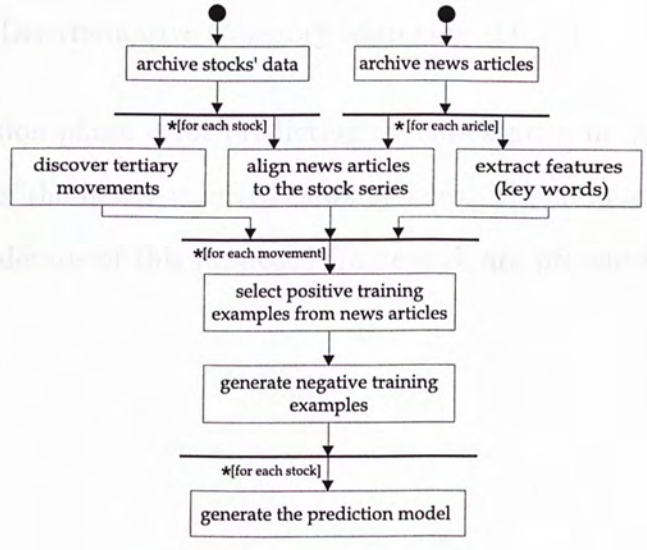
2. **Align News Articles to the Stock Series** News articles are aligned to a stock series based on the broadcasting time of the news articles, as well as with the help of the Efficient Market Hypothesis.

3. **Extract Features (Key Words)** Extracts the main features (key words) from the news articles.

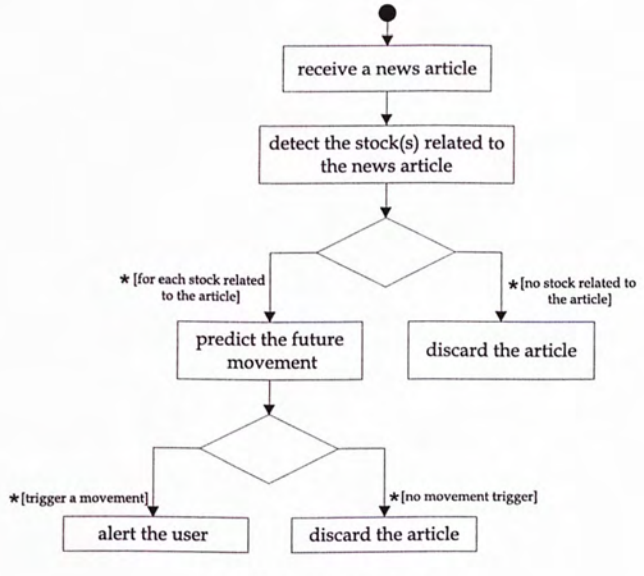
4. **Select Positive Training Examples from News Articles** The selected news articles are believed to trigger or support the tertiary movements on stock prices.

5. **Generate Negative Training Examples** The negative training examples are the news articles that are most contradictary with the positive training examples. They do nothing with the price movements.

[1]Unified Modeling Language™ (UML) – a industry-standard language for specifying, visualizing, constructing and documenting the artifacts of software systems.



(a) Training phase



(b) Operation phase

Figure 2.1: The overview of the system architecture. (a) The training phase. (b) The operation phase.

6. **Generate the Prediction Model** The relationship between the news articles and the stock price movements are learned through a new classification approach, Discriminative Category Matching (DCM).

The operation phase is for predicting the stock price movements according to the contents of the newly received news articles. The unique features and the implementation details of this proposed framework are presented in Part III.

Part II

Literatures Review

Things should be as simple as possible, but not simpler.

Albert Einstein

Chapter 3

The Social Dynamics of Financial Markets

In short, the stock market's fluctuations are the consequence of we, human beings, who bid and ask in the stock market. Investors' decisions on bidding, asking or holding the securities are greatly influenced by what others are saying and doing in the financial market. The emotions of fear, greed, coupled with subjective perceptions and evaluations of the economic condition and their own psychological predispositions and personalities, are the major elements that affect the market behaviors.

Yet, human behaviors are not random. People's actions in the stock market, although occasionally irrational, are predominantly understandable with respect to the social structure, social organization, perceptions and collective beliefs of this complex arena. At times, collective movements become launched, based on a group beliefs about how the market will act or react. In these instances, market trends develop which can be recognized, identified and anticipated to

continue for some periods. Social psychology and sociological insights into human behaviors can be used to supplement economists' expertise in the area of price fluctuations, to provide a more inter-disciplinary understanding of how the stock market operate.

3.1 The Collective Behavior of Groups

People who are involved in financial markets always face a difficult decision among bid, ask and hold. The responsibility for making judgments about the appropriateness of buying and selling is always hazardous, confusing the bold and terrifying the indecisive. A direct result of this situation is the prevailing "sheep mentality", that follow others' thoughts and initiatives, trying to obviate the financial risk and self-deprecation of personal decision making. This anxious conformity within financial markets is very much like a social phenomenon where the peer influences and collective behaviors within groups are studied.

The scientific study of how people generally behave together is regarded as *social psychology*. Social psychology is the study of how thoughts, feelings and behaviors of individuals are influenced by the actual, imagined or implied presence of others. The first attempt to apply psychology to economic was made by a French psychologist, Gabriel Tarde (1843–1904), who has written his first book about psychology economic, *La Psychologie Economique*, in 1902 [55]. Tarde was followed by another psychologist, John Maurice Clark (1884–1963), who published his first article related to economics psychology, *Economics and Modern Psychology*, in the *Journal of Political Economy* in 1918 [9]. The idea of economic psychology immediately became an essential element in European economics and

soon spread worldwide. Different theories originated from psychology and sociology are then proposed and developed to account for the price fluctuations due to people's fear of making a wrong decision of incurring a loss or missing a profit.

In general, when the unpredictable vagaries of the market challenge the beliefs of the investors, *cognitive dissonance* will occur. Cognitive dissonance theory was proposed by an American psychologist, Leon Festinger (1920–1989) who contends that a person will experience feelings of psychological stress when he/she holds some *cognitions* that are inconsistent with the others [18]. Cognitions refer to ideas, thoughts, beliefs or opinions. A typical example of cognition inconsistence is like this: an investor belief that the price of a security will increase while its price is actually dropping. Since the holding of incompatible cognitions produces feelings of unease and tension, one becomes motivated to reduce or eliminate these emotions by using two major strategies: 1) Change some of the cognitions so that they will be consistent with the others; or 2) Add additional cognitions so as to make the inconsistent cognitions become compatible [28]. In order to have some good references for modifying the present cognitions, one will often react by engaging in *social comparison.*

Social comparison theory is again proposed by Festinger, who stated that humans have a relentless desire to evaluate their opinions, beliefs and attitudes by comparing their views with each other [17]. An action would be perceived as correct, valid and proper to the degree that it is anchored in a group of people with similar behaviors, provided that objective measurement is unavailable or inapplicable to determine the appropriateness of the action. In the case of financial market, when the price of a security is dropping, there is no objective measurement to indicate that whether this dropping is just an instant or is a

trend. Spurred by fears and doubts, joys and beliefs, and surrounded by others in a similar state, coupled with the social comparison process, investors mainly follow the people who are thought to be successful or have superior knowledge and power, such as market gurus or a group of mysterious defined as "The market" [2]. As a result, investors are highly susceptible to *social contagion.*

Social contagion theory was introduced by a French psychologist, Gustav LeBon (1841–1931) who attempted to analyze and explain crowd behavior in his famous book *La Psychologie des foules* [31]. Social contagion theory states that increased degrees of suggestibility, emotionality and emotional diffusion lead members of a group to behave in similar ways [54]. One of the most fundamental elements from the social contagion theory is *circular reaction.* Circular reaction refers to a form of inter-stimulation where the behavior of individual serves to stimulate other individuals to produce the same behavior, which is then reflected back to those who initiated the behavior [6]. As a result, there is an escalating degree of emotionality and similar behavior, each one reinforcing and intensifying the others. The widespread, immediate and readily available news about the financial markets and the economic conditions enable the circular reaction to occur, and hence facilitate the social contagion process. For instance, a group of investors may follow the forecasting result from a market analyst who predicts the market would go down. Influenced by this group of investors, the market goes down, and the analyst further assure that his prediction is correct and further release unpresent news in the market.

3.2 Prediction Based on Publicity Information

The first and the most systematic examination against the impact of textual information on the financial markets was conducted by Klein and Prestbo [29] in 1974. Their survey consisted primarily of a comparison of the movements of Dow Jones Industrial Average[1] with general news during the period from 1966 to 1972. The news stories that they have taken into consideration were the materials appearing in the "What's New" section of the *Wall Street Journal*, as well as three feature stories[2] carried on the Journal's front page. The major criticism of their study is that too few news stories were taken into consideration in each day. It is rather simple to assume that stories carried on the front page of a newspaper are enough for summarizing and reflecting the information appear in the whole newspaper. Interestingly, even with such a simple setting, Klein and Prestbo found that the pattern of directional correspondence, whether upwards or downwards, between the flow of the news and stock price movements manifested itself 80% of the time. Their findings strongly suggested that news and market tend to move together.

Fawcett and Provost [16] formulate an *activity monitoring task* for predicting the stock price movements based on the content of the news articles. Activity monitor task is defined as the problem that involves monitoring the behaviors of a large population of entities for interesting events which require actions. The objective of the activity monitoring task is to issue alarms accurately and quickly. In stock price movements detection, news articles and stock prices for approx-

[1] A financial index which composed of 30 blue-chip stocks listed on the New York Stock Exchange.

[2] Klein and Prestbo did not describe in details how they selected these three stories among all stories carried on the Journal's front page.

imately 6000 companies over three months period are archived. An interesting event is defined to be a 10% change in stock price which can trigger by the content of the news articles. The goal is to minimize the number of false alarms and to maximum the number of correctly predicted price spikes. It is worth noting that, the authors only provides a brief framework for formulating this predicting problem. The implementation details and an in-depth analysis are both missing. Perhaps this is because their main focus is not on examining the possibility of detecting stock price movements based on news stories, but is on outlining a general framework for formulating and evaluating the problems which require continuous monitoring their performance.

Thomas and Sycara [53] predict the stock prices by integrating the textual information that are downloaded from web bulletin boards[3] into the trading rules which are derived by genetic algorithms based on numerical data. For the textual data, a maximum entropy text classification approach [41] is used for classifying the impacts[4] of the posted messages on the stock prices. For the trading rules, they are constructed by genetic algorithms based on the trading volumes of the stocks concerned, as well as the number of messages and words posted on the web bulletin boards per day. A simple market simulation is conducted, and they reported that the profits obtained increased up to 30% by integrating the two approaches than using either of them. However, the technical strength of their paper is too weak and the framework is too simple, such that no analysis on their results was given.

Wuthrich et al. [62] develop an online system for predicting the opening

[3]Thomas and Sycara chose Forty discussion boards from `www.ragingbull.com`
[4]Two impacts are defined in their paper: up and down.

prices of five stock indices[5] by analyzing to the contents of the electronic articles downloaded from the *Wall Street Journal* [6]. The analysis is done as follows: for each article, keywords are extracted and weights are assigned to them according to their significance in the corresponding piece of news article and on the corresponding day. By combining the weights of the keywords and the historical closing prices of a particular index, some probabilistic rules are generated by using the approach proposed by Wuthrich [60, 61]. Based on these probabilistic rules, predictions which indicated that there would be at least 0.5% price changes are made. The major weaknesses of their system is that only the opening prices of financial markets could be predicted. Some others more challenging and interesting issues, such as intra-day stock price predictions, could not be achieved.

Following the techniques proposed by Wuthrick et al., Permunetilleke and Wong [43] repeat the work but with different domain. News headlines (instead of news contents) are used to forecast the intra-day currency exchange rate (instead of opening prices of stock indices). These news headlines belong to world financial markets, political or general economic news. They show that on a publicly available commercial data set, the system produces results are significantly better than random prediction.

Lavrenko et al. [30] propose a system, Analyst, for predicting the intra-day stock price movements by analyzing the contents of the real-time news articles. Analyst is developed based on a language modeling approach proposed by Ponte and Croft [45]. While a detailed architecture and a fruitful discussion are both presented in their paper, some questions still left unanswered. The authors claim

[5]These five stock indices are: Dow Jones Industrial Average, Nikkei 225, Financial Times 100 Index, Hang Seng Index and Singapore Straits Index

[6]`http://www.wsj.com`

that there should be a period, t, to denote the time for the market to absorb any new information (news articles) release, where t is defined as five hours. We have to admit that the market would spent time to digest an information. However, such a period would be too long that contradict with most economic theories. In addition, the derivation of t seems to be trial-and-error that lacks of theoretical foundation. Furthermore, a piece of news article may frequently classify to trigger both the rise and drop movements of the stock prices in the training stage, which is clearly a dilemma. Finally, in their evaluation, even a prediction is incorrect, a profit could still be obtained.

Chapter 4

Time Series Representation

As with most data mining problems, data representation is the key to efficient and effective solution. A sound time series representation involves issues such as recognizing the significant movements or detecting any abnormal behaviors, so as to study and understand its underlying structure. Over centuries, thousands of high-level time series representation techniques have been proposed. Technical analysis, which is frequently used among financial analysts, and piecewise linear approximation, which is mostly used in the academic community, are two popular and promising approaches for representing and discovering patterns on the financial time series. This chapter briefly reviews these two techniques.

4.1 Technical Analysis

The root of technical analysis could trace back to the Dow theory, which is originated from Charles Henry Dow (1850–1902) [5, 15, 39]. Dow had published a series of articles on the mechanisms in stock price movements from 1900 in the

Wall Street Journal. His primary interest was using stock market analyses to forecast the economy, but in practice stock-dealers are far more interested in his theories than economists.[1] Unfortunately, he had never published any material on his observations other than the aforementioned newspaper articles before he dead. The main concepts from these newspaper articles were later summarized by an acquaintance, Samuel A. Nelson, in his book *The ABC of Stock Market Speculation* [39], into what we today call Dow Theory. Dow Theory comprises the following six observations:

1. **Stock indices discount every information in advance** The information, that the market participants hold for interpreting the importance of a stock value, is discounted in the markets price fluctuations.

2. **Stock price has three movements: primary, secondary and tertiary** Primary, secondary and tertiary movements last for years, months and weeks, respectively. They act simultaneously but not necessarily in the same direction.

3. **Support and resistance areas give price signals** When prices fluctuate within a given range, the lower and upper end are called support and resistance level, respectively. Whenever the price rises above the resistance level (or falls below the support level), a break-out occurs, which indicates a rising (or dropping) trend begins.

4. **Volume supports price** The significant of a price movement as a signal is strengthened if it appears at a high volume. A price movement associated

[1]The Dow Jones Industrial Average is one of the direct result of his methods of analyzing the financial markets.

with a light volume is considered as a temporary move only.

5. **A trend must be confirmed by peaks and troughs** As long as all new peaks and troughs are rising (or dropping), a rising trend (or dropping trend) is intact.

6. **Both the Industrial Index and Transportation Index must confirm a stock trend.** An initial trend in a stock market is not credible until both the stated Indices have started the trend movement.

By reviewing and modifying these observations, numerous books and articles are published. For those techniques which extend the ideas from the above observations are collectively regarded as technical analysis. Nowadays, most of the techniques developed in technical analysis are based on the classic book *Technical Analysis of Stock Trends*, written by Robert D. Edwards and John Agee [15]. It explains how technical analysis is based on the following assumptions [5]:

1. Market value is determined solely by the interaction of demand and supply.

2. Stock prices tend to move in trends that persist for long periods of time.

3. Shifts in demand and supply cause reversal in trends, and can be detected by charts.

4. Many chart patterns tend to repeat themselves.

Thousands of Technical Indicators for prediction of stock prices have been developed, and are still developing on the ground of these basic observations and assumptions. A thorough survey of the most common technical indicators can be found in *Technical Analysis from A to Z* by Steven B. Achelis [1].

4.2 Piecewise Linear Approximation

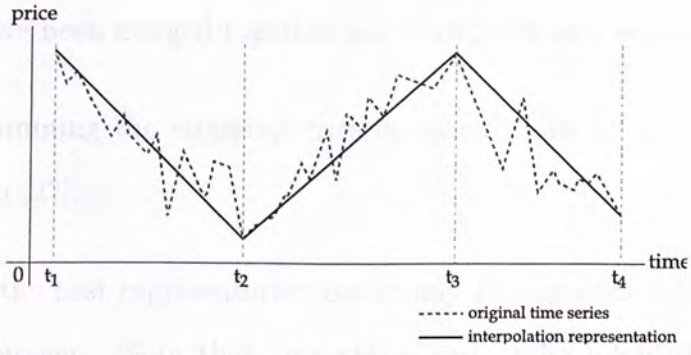
Piecewise linear approximation, or sometimes called piecewise linear segmentation, refers to the idea of representing a time series of length n using K straight lines, where $K \ll n$ in most cases [27, 42]. It is one of the most widely used technique for high-level time series representation, especially for the financial time series [30, 46, 59]. Most studies in this area are pioneered by Pavlidis et al. [42] and Dedua and Harts [13].

Given a time series, the straight lines that approximate it can either be *linear interpolation* or *linear regression* [27]. Figure 4.2 illustrates their differences. Linear interpolation approximation (Figure 4.2 (a)) refers to closely align the endpoints of consecutive segments, such that a smooth-look upon the segmentation result is obtained. In contrast, linear regression (Figure 4.2 (b)) may produce a disjoint-look on the same time series. In general, linear interpolation consumes lower computational resources, while linear regression gives a higher quality in terms of Euclidean distance [27].

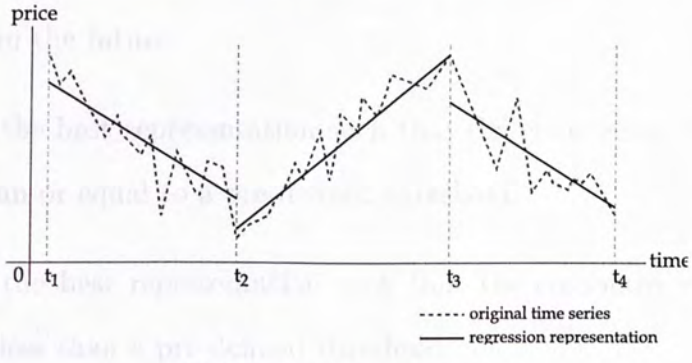
Broadly speaking, most of the segmentation algorithms can be grouped into one of the following three categories:

1. **Sliding-Window** A straight line is trying to approximate the data points on a time series from time t_i up to time t_j ($t_i < t_j$), where some stopping criteria are met at t_j. The sub-sequence of the time series from t_i to t_j is transformed into one segment. The next segment immediate begins at the point where the previous segment ended, and the whole process continues recursively until the end of the time series.

2. **Top-Down** It works by considering every data point on the time series



(a) Interpolation representation



(b) Regression representation

Figure 4.1: Different kinds of piecewise linear segmentation to represent a time series. (a) Interpolation representation. It gives the time series a smooth-look. (b) Regression representation. It gives the time series a disjoint-look.

as a potential partitioning point, and recursively partition the time series into two sub-sequence until some stopping criteria are met. Finally, each partition left is regarded as a distinct segment.

3. **Bottom-Up** It is a natural complement to the top-down segmentation approach. Initially, each data point on the time series is regarded as a potential merging point. Starting from the finest possible approximation,

the data points are merged until some stopping criteria are met. The points which have been merged together are considered as a segment.

For determining the stopping criteria, usually one of the following three policy is chosen [27]:

1. Produce the best representation using only K segments, where K is a pre-defined integer. Note that this policy cannot be adopted in the online segmentation mode as it is impossible to know exactly how the time series behaves in the future.

2. Produce the best representation such that the error norm of each segment is less than or equal to a pre-defined threshold.

3. Produce the best representation such that the combined error of all segments is less than a pre-defined threshold.

Among the three segmentation approaches, sliding-window is an attractive one as it is simple, intuitive and particularly because it could be applied online. Furthermore, it performs quite well upon some noisy data, especially upon financial data [46, 59]. However, when comparing with the other two approaches, the performance of sliding-window is far left behind [27]. For the top-down and bottom-up approach, they can only be operated in offline mode, and they require to scan the entire data set. This makes them impractical or even be infeasible in a data-mining context where the data are in the order of terabytes or arrive in continuous streams. Keogh et al. [27] modify the bottom-up segmentation approach and let it be able to apply online. However, their proposed algorithm results in

a time lag which may not be feasible to use in the circumstance where decisions have to be made very quickly, such as medical diagnose or stock exchange.

Chapter 5

Text Classification

Text classification is a supervised learning task, defined as labeling natural language texts with thematic categories from a pre-defined set. It dates back to the early '60s, but only in the early '90s did it become a major sub-field of the information systems related discipline.

Until the late '80s, the most popular approach was the knowledge engineering one, which consists of manually defining a set of rules encoding expert knowledge on how to classify documents under some given categories. In the '90s, this approach lost popularity, and text classification was in favor of the machine learning paradigm, according to which a general inductive process automatically builds a classifier by learning the characteristics of the categories of interest from a set of pre-classified documents.

The advantages of machine learning approach over the knowledge engineering approach include very good classification effectiveness, considerable savings in terms of expert labor power, and straightforward portability to different domains. This chapter reviews the main approaches to text classification that fall within the

machine learning paradigm. Three issues are discussed in details: 1) Document representation; 2) Document pre-processing; and 3) Classifier construction.

5.1 Document Representation

The choice of text representation depends on what one regards as meaningful units in text (the problem of lexical semantics) and meaningful natural language rules for the combination of these units (the problem of compositional semantics). In text classification scenario, the later problem is usually disregarded [51]. A document, d_i, is usually represented as a vector of *features* and *weights*:

$$d_i = \langle (f_1, w_1), (f_2, w_2), \ldots, (f_n, w_n) \rangle \tag{5.1}$$

where f_j is a feature appears in the document collection, and $0 \leq w_j \leq 1$ represents how significant f_j contributes to the document d_i.

A feature usually denotes a single word. In a number of researches [3, 14, 32], it has been shown that representations that are more sophisticated than Equation (5.1) do not yield any significant improvement in terms of classification quality. Some authors use phrases, rather than individual words, as indexing terms [21, 50, 56], but the results obtained are discouraged, irrespective of how the phrases are motivated[1]. Quite convincingly, Lewis [32] suggests that the most likely reason for the discouraging results is that although indexing languages based on phrases have superior semantic qualities, they have inferior statistical qualities

[1]Two kinds of motivation exist: linguistics and statistics. The former one bases on grammatical structure, while the latest one composes of a sequence of words that occur contiguously with high frequency in the collection.

with respect to the single word indexing technique.

For determining the weight w_j of feature f_j in document d_i, usually one of the following ways is chosen, depending on what type of classifier[2] one intends to use:

- Assign a boolean value to the weights, such that $w_j \in \{0, 1\}$ indicating that whether or not the feature f_j appears in document d_i.
- Formulate the weights such that they are as same as the number of times they appear in document d_i.
- Apply a weighting scheme to calculate the weights. The most widely used one is the $tf \cdot idf$ scheme proposed by Salton [49]:

$$tf \cdot idf(f_j, d_i) = tf(f_j, d_i) \cdot \log \frac{N}{df(f_j)} \tag{5.2}$$

where $tf(f_j, d_i)$ is the number of times the feature f_j appears in the document d_i; $df(f_j)$ is the number of documents contain the feature f_j; and N is the total number of documents in the entire collection. This function encodes two intuitions: 1) the more often a feature appears in a document, the more representative of the document's content it is; and 2) the more documents contain the feature, the less discriminative it is. In order to account for the differences in document length, the weights calculated are often normalized into unit value. Note that if the feature distribution is changed, we require to re-calculate and re-assign the weights to every

[2]Details of the type of classifier would be discussed in following section.

feature in each document from scartch. This comsumes very much computational resources.

5.2 Document Pre-processing

Very often, before indexing, the removal of *functional words* and *stemming* are always performed. Function words are defined as those topic-neutral words, such as prepositions, conjunctions, digits, etc [51]. Stemming is used to improve the retrieval effectiveness and to reduce the size of indexing files by relating morphologically similar indexing and searching terms [19].

Besides, the high dimensionality of the feature space is problematic. Most of the well-formulated learning algorithms in classification cannot scale well. Hence, before constructing the classifier, one often applies a pass of *dimensionality reduction*, which aims at reducing the size of the feature space. Dimensionality reduction is usually achieved by applying *feature selection*. Two kinds of feature selection techniques exist, namely, 1) thresholding document frequency; and 2) applying selection function.

For the former technique, only the features that appear in more than M documents are retained, where M is a predefined number. Yang and Pedersen [65] show that this techniques can reduce the feature dimension by a factor of 10 without any loss in classification quality. Furthermore, a reduction by a factor of 100 only brings a small loss.

For the latest technique, various methods are developed based on information theory or probabilistic foundation. Among them, χ^2 [7, 65], information gain [7, 65], mutual information [47, 65], NGL coefficient [40, 47] and odd ration [7, 47]

are some of the most popular selection functions. In short, these functions try to capture the intuition that the best features are those which distribute most differently in the sets of positive and negative examples. However, the interpretations of this principle vary across different functions. In general, the experiments reported in the aforementioned papers seem to indicate that NGL coefficient and odd ratio perform the best while mutual information performs the worst.

Finally, it is worth noting that dimensionality reduction by applying selection function needs to process the entire collection from scratch whenever we need to update the classifier. This is very expensive especially for the case where the classifier requires to update frequently.

5.3 Classifier Construction

Text classifier construction has been tackled, and still tackling, in a variety of different ways. Here, I will describe in details only the methods that have been proven to be the most effective and popular in the recent literatures. Thus, two classifiers are discussed: Naive Bayes and support vectors machine. Among all approaches, Naive Bayes and support vectors machine perform the best in term of efficiency and effectiveness, respectively.

5.3.1 Naive Bayes (NB)

NB is a probabilistic based algorithm in which its basic idea is to compute the posterior probability of the incoming document given a particular category [32, 34]. By applying the rules derived by Reverend Thomas Bayes (1702–1761), this

posterior probability becomes:

$$P(d_i|c_k) = \frac{P(c_k) \cdot P(d_i|c_k)}{P(d_i)} \tag{5.3}$$

where c_k is one of the pre-defined categories; $P(d_i)$ is the probability that a randomly picked document has vector d_i as its representation; and $P(c_k)$ is the probability that a randomly picked document belongs to the category c_k.

The estimation of $P(d_i|c_k)$ is extremely difficult, since the number of possible vectors for document d_i is too many. In order to alleviate this problem, it is common to make the assumption that the occurrence of every feature is statistically independent of each other, such that $P(f_a) = P(f_a|f_b)$, where f_a and f_b are two different features. As a result, the classifier is "naive" as this assumption is usually not verified in practice [32, 51].

Broadly speaking, two versions of NB exist, the Multivariate Bernoulli model and the Multinomial mixture model. For the document representation, the former one only takes care of whether or not a particular feature appears in the document, while the latest one further considers how many times a feature appears in each document. Recent researches indicate that the multinomial mixture model performs superior than the multivariate Bernoulli model[3] [4, 35]. An efficient implementation of NB is the Rainbow package developed by McCallum[4].

The merit of NB lies on its robustness. It consumes very little computational resources. Its training cost is linear with the size of the training data while its operation cost is also very low [36]. Furthermore, it is possible to update the

[3]However, these studies conducted only on a subset of some benchmark data, and are unable to compare with those previous studies directly

[4]http://www.cs.cmu.edu/~mccallum/bow

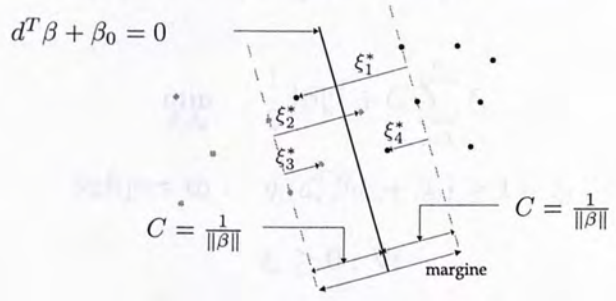


Figure 5.1: The basic concept of a support vectors classifier in a two-dimensional situation. The decision boundary is the solid line, while the broken lines bound the maximal margin of width $2C$. The points labeled ξ_j^* are on the wrong side of their margin by an amount $\xi_j^* = C\xi_j$. Points on the correct side have $\xi_j^* = 0$.

model continuously and incrementally as it only stores the probabilities of the features appearing in each category [51].

It is worth noting that NB requires significant text preprocessing, such as feature selection [35, 64], or else its quality will be poor. To make it worse, even with feature selection, its quality is not high when comparing with other classification approaches [36, 64]. Although the quality of NB can be improved by maintening some association-like rules [20, 36, 37], the computational cost for both training and operation would be far more expensive than before.

5.3.2 Support Vectors Machine (SVM)

SVM is a learning algorithm proposed by Vapnik based on structural risk minimization principle for solving two-class pattern recognition problem [58]. Conceptually, it tries to generate a decision hyperplane that maximizes the margin between the positive and the negative examples in a training set. Figure 5.1 illustrates this concept using a two dimensional situation. Mathematically, it requires

to solve the following quadratic programming problem [10, 23, 58]:

$$\min_{\beta,\beta_0} : \quad \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i \tag{5.4}$$

$$\text{subject to} : \quad y_i(d_i^T\beta d_i + \beta_0) \geq 1 - \xi_i \ , \ \forall i \tag{5.5}$$

$$\xi_i \geq 0 \ , \ \forall i \tag{5.6}$$

where d_i is a training document; $y_i \in \{-1, 1\}$ indicates to which side the document d_i belongs; C is a constant; and n is the size of the training data. For computational reason, it is far more efficient to solve the above primal optimization problem by converting it to the Lagrangian (Wolfe) dual optimization problem [10, 23, 58]:

$$\min : \quad L = -\sum_{i=1}^{n}\alpha_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n}\alpha_i\alpha_{i'}y_iy_{i'}d_i^Td_{i'} \tag{5.7}$$

$$\text{subject to} : \quad \sum_{i=1}^{n}\alpha_iy_i = 0 \tag{5.8}$$

$$0 \leq \alpha_i \leq C \ , \ \forall i \tag{5.9}$$

The size of the optimization problem depends on the number of training data n. Define a matrix $Q = y_iy_{i'}d_i^Td_{i'}$. The size of Q is around n^2. For learning task consists of thousand of features and thousand of training documents, Q becomes impossible to keep in memory. Standard implementations require either explicit storage of Q or re-compute Q every time when it is needed. However, this becomes prohibitively expensive. Joachims [25] proposes a promising solution by decomposing the learning task into several sub-tasks, which could achieve a nearly optimal solution.

The above algorithm is used for solving linearly separable case, but it can also be extended to solve linearly non-separable case by either introducing some soft margin hyperplans or mapping the original document vectors to a higher dimensional space [10, 23, 58]. Recent researches show that SVM achieves very high quality for text classification, which outperforms all of the existing approaches [8, 26, 51, 64]. Some efficient implementation of SVM include SVM^{light} by Joachims[5] [25] and SMO by Platt[6] [10, 44].

The major shortcoming of SVM is that its training cost is extremely expensive. In fact, its training time is quadratic to the number of training instances [8, 10]. In addition, SVM requires feature weighting, such as applying $tf \cdot idf$ scheme [26, 64], or else its performance degrades significantly. These two issues lead SVM becomes difficult in dealing with the situation where the classifier needs to update frequently.

[5]http://svmlight.joachims.org/
[6]http://research.microsoft.com/~jplatt/smo.html

Part III

Mining Financial Time Series and Textual Documents Concurrently

Great God! This is an awful place!

Sir Robert Falcon Scott

Chapter 6

Time Series Representation

For any analysis related to financial time series, a high-level re-describing or segmenting the time series into trends is always necessary. This is because such kind of time series is always very noisy. Numerous approaches exist to achieve this goal. Technical analysis is properly the easiest way to decomposite a time series into trends. Piecewise linear approximation is another popular technique to do so. In this chapter, a novel t-test based split and merge segmentation algorithm, a kind of piecewise linear approximation technique, is proposed. Two phrases are included in this algorithm, splitting phrase and merging phrase. The splitting phrase aims at discovering trends on the time series, while the merging phrase aims at avoiding over-segmentation.

6.1 Discovering Trends on the Time Series

From a stock trader point of view, general trends of the stock price movements are far more informative than the exact price fluctuations as our decisions on

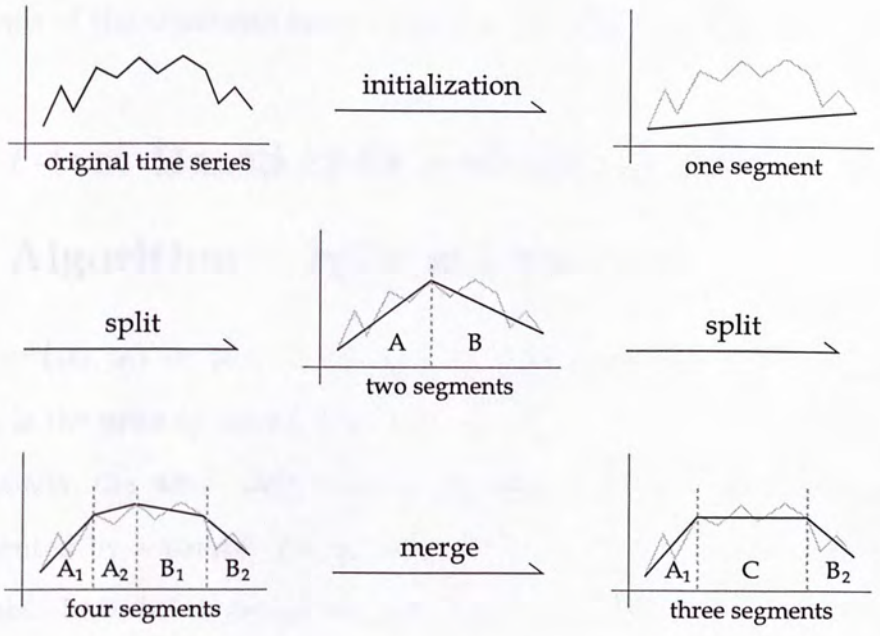


Figure 6.1: General idea of the t-test based split and merge segmentation algorithm. The splitting phase aims at discovering all of the possible trends on the time series, while the merging phase aims at avoiding over-segmentation.

bidding or asking are based on the recent trends but not on just a few previous observations. As a result, a high-level time series representation is properly the first and the most critical procedure.

In order to decomposite a time series into trends, a t-test based split and merge segmentation algorithm, which belongs to the family of interpolation of piecewise linear approximation, is proposed. This segmentation algorithm consists of top-down segmentation approach and bottom-up segmentation approach.[1]

[1]Please refer to Chapter 4 Section 4.2 for the meaning of top-down and the bottom-up segmentation approaches.

Figure 6.1 illustrates the general idea of the proposed segmentation algorithm. The details of the algorithm are given in the following sections.

6.2 t-test Based Split and Merge Segmentation Algorithm – Splitting Phrase

Let $T_n = \{(t_0, p_0), (t_1, p_1), \ldots, (t_n, p_n)\}$ be a financial time series of length n, where p_i is the price at time t_i for $i \in [0..n]$.

Initially, the whole time series is regarded as a single large segment, and is represented by a straight line joining the first and the last data points of the time series. In order to decide whether this straight line (segment) is adequate to denote the general trend of the time series, a one tail t-test is formulated:

$$\begin{aligned} H_0 &: \varepsilon = 0 \\ H_1 &: \varepsilon > 0 \end{aligned} \tag{6.1}$$

where ε is the expected mean square error of the straight line with respect to the actual fluctuation on the time series. It is computed by:

$$\varepsilon = \frac{1}{n} \cdot \sum_{i=0}^{n} (p_i - \hat{p}_i)^2 \tag{6.2}$$

where n is the total number of data points within the segment, $\hat{p}_i$ is the projected price of p_i for the given time t_i. The required t-statistics is:

$$t = \frac{\varepsilon}{\sqrt{\hat{\sigma}^2/n}} \tag{6.3}$$

Algorithm 1 split($T_n[t_a, t_b]$) – split a time series T_n of length n from time t_a to time t_b where $0 \leq a < b \leq n$

```
1: ε_min = ∞;
2: ε_total = 0;
3: for i = a to b do
4:    ε_i = (p_i − p̂_i)^2;
5:    if ε_min > ε_i then
6:       ε_min = ε_i;
7:       k = i;
8:    end if
9:    ε_total = ε_total + ε_i;
10: end for
11: ε = ε_total/(t_b − t_a);
12: if t-test.reject(ε) then
13:    split(T_n[t_a, t_k]);
14:    split(T_n[t_k, t_b]);
15: end if
```

where $\hat{\sigma}$ is the standard deviation of the mean square error, ε. The t-statistics is compared with the t-distribution with $n - 1$ degree of freedom using $\alpha = 0.05$. In other words, there is a probability of 0.05 that the null hypothesis (Equation (6.1)) would be accepted given that it is incorrect.

If the null hypothesis is rejected, then the straight line (trend) is split at the point where the error norm is maximum, i.e. $\varepsilon_{max} = \max\{(p_i - \hat{p}_i)^2\}$ for $i \in [0..n]$. The whole process will be executed recursively on each segment until all of the segments are accepted by the t-test. Algorithm 1 outlines the procedure of the splitting phrase.

Algorithm 2 merge($T'_m[t'_a : t'_b]$) – attempt to merge two adjacent segments on the time series T'_m of length m from t'_a to t'_b where $0 \leq a < b \leq m$

```
1: while true do
2:   ε_min = ∞;
3:   for i = a to b do
4:     ε_total = 0;
5:     ε_i = Σ_{j=t'_i}^{t'_{i+2}} (p_j − p̂_j)^2;
6:     if ε_min > ε_i then
7:       ε_min = ε_i;
8:       k = i + 1;
9:     end if
10:  end for
11:  ε = ε_min/(t'_{k+1} − t'_{k−1});
12:  if t-test.accept(ε) then
13:    drop (t'_k, p'_k) from T'_m;
14:  else
15:    break;
16:  end if
17: end while
```

6.3 t-test Based Split and Merge Segmentation Algorithm – Merging Phrase

After the splitting phrase, *over-segmentation* may frequently occurred. Over-segmentation refers to the situation where there exist two adjacent segments such that their slopes are similar, and they could be merged as a single large segment instead of splitting into two distinct segments. Let us refer to Figure 6.1 again. If we only perform the splitting phase, four segments would be resulted. However, note that the slopes of segment A_2 and segment B_1 are very similar. Hence, merging of them is therefore possible. After merging A_2 and B_1, three segments will be reminded, such that they all have different slopes. In other words, merging

phase aims at combining all adjacent segments, provided that the mean square error, ε, would still be accepted by the t-test after merging. The hypothesis for the t-test is as same as Equation (6.1).

Formally, consider the time series T_n which has been transformed into another time series $T'_m = \{(t'_0, p'_0), (t'_1, p'_1), \ldots, (t'_m, p'_m)\}$ of length m after the splitting phase, such that $m < n$. If the null hypothesis over two adjacent segments, $\{(t'_i, p'_i)(t'_{i+1}, p'_{i+1})\}$ and $\{(t'_{i+1}, p'_{i+1})(t'_{i+2}, p'_{i+2})\}$, is accepted, then these two segments are regarded as a *candidate merging pair*. Let $\mathcal{L}_{merge}$ be a list containing all of these candidate merging pairs. One of the candidate merging pair resides in $\mathcal{L}_{merge}$ would be selected to merge if merging of it would result in the minimum increase in the total error norm. The whole process is executed continuously until the t-test over all of the segments on the time series is rejected, i.e. $\mathcal{L}_{merge} = \phi$. Algorithm 2 illustrates the whole procedure.

Chapter 7

Article Alignment and Pre-processing

In order to discover the relationship between news articles and price movements, article alignment is done. Article alignment is the process of aligning news articles to the stock trends, such that the aligned articles could trigger or support these trend movements.

Different scholars may have different interpretations about how the alignment process should be done. It is difficult, if not impossible, to find a completely consensus. Some may favor the idea of having a time lag between the news article broadcast and the price movement, so as to denote the time that the market spends for absorbing the new information. However, according to the Efficient Market Hypothesis, a long time lag is normally impossible. This chapter focuses on two issues. The first issue is how the news articles is aligned to the stock trends. The second issue is how the useful news articles that would trigger or support the price movements are selected .

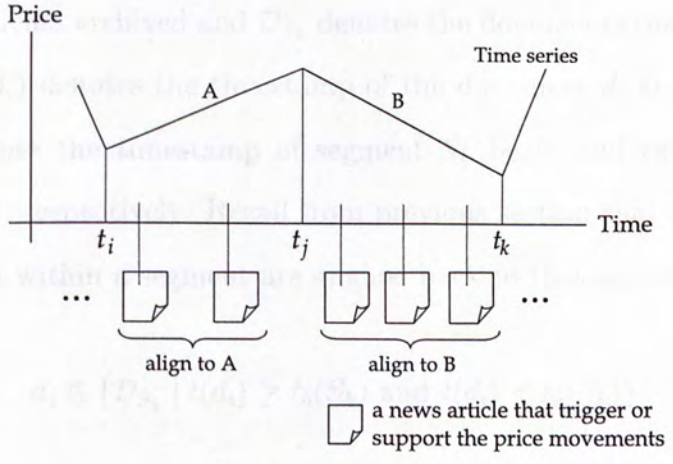
7.1 Aligning News Articles to the Stock Trends

After the time series segmentation process, stock trends are generated. New articles could then align to these trends such that the aligned news articles are highly believed to trigger or support the stock trend movements. Perhaps the simplex way of article alignment is to apply expert knowledge, such that we manually classify the impact of each piece of news article on the stock movements. But obviously, this approach is rather inefficient. There are too many news articles and stocks in the nation's financial market[1], such that it is impossible for us to read through all of them preciously.

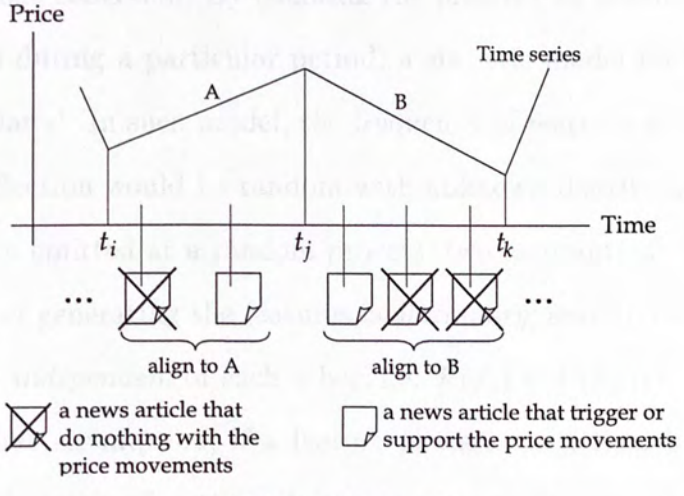
Thanks to the strong form of the Efficient Market Hypothesis (EMH), which states that the market is an efficient information processor that would reflect the assimilation of all of the information available immediately, we therefore associate the relationship between the news articles to the stock trends in the way such that if a news article is broadcast as time t_i, the stock price will adjust itself at t_i immediately thereafter. Figure 7.1 (a) illustrates this idea. In Figure 7.1 (a), the news articles which are broadcasted within the time t_i to t_j are responsible for triggering or supporting the rise movement, while those articles broadcasted within the time t_j to t_k are responsible for the drop movement.

Unfortunately, not every piece of news article would be responsible for triggering or supporting the price movements. Figure 7.1 (b) illustrates this concept. As a result, a selecting heuristics is necessary for selecting those valuable news articles. This heuristics is separated into two procedures, selecting positive training articles and selecting negative training articles.

[1]Take Reuters as an example, around 900,000 news articles broadcasted within a year.



(a) An ideal situation.



(b) The truth situation.

Figure 7.1: The basic idea of the alignment process. (a) An ideal situation where all of the news articles broadcasted under a stock trend are responsible for triggering or supporting the movement of the corresponding trend. (b) In reality, some news articles do nothing with the price movements.

7.2 Selecting Positive Training Examples

Define S_k to be a segment and d_i to be a document (news article). Let $\mathcal{D}$ denotes all of the documents archived and $\mathcal{D}_{S_k}$ denotes the documents that are aligned to segment S_k; $t(d_i)$ denotes the timestamp of the document d_i announced; $t_b(S_k)$ and $t_e(S_k)$ denote the timestamp of segment S_k begin and the timestamp of segment S_k end, respectively. Recall from previous section that documents that are broadcasted within a segment are aligned back to that segment, i.e.:

$$d_i \in \{\mathcal{D}_{S_k} \mid t(d_i) \geq t_b(S_k) \text{ and } t(d_i) < t_e(S_k)\} \tag{7.1}$$

Define *features* to be any words in the news article collection. Let f_j be a feature in the news article collection. By counting the presence or absence of a specified feature appears during a particular period, a statistic model for discrete events could be formulated. In such model, the frequency of features appear within the news article collection would be random with unknown distribution. In a model that features are emitted at a random process, two assumptions could be made: 1) The process of generating the features is *stationary*; and 2) The occurrence of every feature is *independent* of each other, i.e. $P(f_a) = P(f_a|f_b)$.

For the first assumption, if a feature is stationary, then in any arbitrary period, the probability of getting it is the same as at any other periods. In other words, if the probability of a feature appearing in some periods change dramatically, we can conclude that this feature exhibit an abnormal behavior in those periods, and it would be regarded as an important feature in there. Specifically, by counting the number of documents that: 1) contains feature f_j and in Segment S_k; 2) contains feature f_j but not in Segment S_k; 3) does not

	#documents contain f_j	#documents do not contain f_j
Segment $= S_k$	case 1	case 3
Segment $\neq S_k$	case 2	case 4

Table 7.1: A 2×2 contingency table summarized the distribution of feature f_j in the document collection. This table could be modeled by a χ^2 distribution with one degree of freedom.

contain feature f_j but in segment S_k; and 4) does not contain feature f_j and in segment S_k, a 2×2 contingency table could be formulated (Table 7.1). Note that this table could be modeled by a χ^2 distribution with one degree of freedom.

For the second assumption, it is known as independent assumption of feature distribution, which is nevertheless a common assumption in text information management, especially for information retrieval, clustering and classification. It seems contradictory to our knowledge. Interestingly, researches show that this assumption will not harm the system performance [12, 33, 34, 51]. Instead, maintaining the dependency of features is not only extremely difficult, but also may easily degrade the system performance if done carelessly [11, 34, 51, 57].

For each feature f_j under each segment S_k, we calculate its χ^2 value, i.e. $\chi^2(f_j, S_k)$. If it is above a threshold, α, i.e. $\chi^2(f_j, S_k) \geq \alpha$, we conclude that the occurrence of feature f_j in segment S_k is significant, and this feature is appended into a feature list, $\mathcal{L}_{feature,S_k}$. $\mathcal{L}_{feature,S_k}$ stores all the features in which their occurrence in segment S_k are significant.

Define $\mathcal{D}_{R,+}$ and $\mathcal{D}_{D,+}$ be two sets containing the documents that support the rise movement and drop movement, respectively. In other words, these two sets are served as the positive training examples. A document, d_i, which belongs to segment S_k ($d_i \in \mathcal{D}_{S_k}$), would be assigned to $\mathcal{D}_{R,+}$ if and only if the slope of

S_k is positive and d_i contains a feature listed in $\mathcal{L}_{feature,S_k}$ (i.e. $d_i \in \mathcal{D}_{R,+}$ iff $f_j \in \{\mathcal{L}_{feature,S_k}$ and $d_i \text{where} d_i \in \mathcal{D}_{S_k}\}$). Similar strategy applies to $\mathcal{D}_{D,+}$.

Note that for $\chi^2 = 7.879$, there is only a probability of 0.005 that a wrong decision would be made such that a feature from a stationary process would be identified as not stationary, i.e. a random feature is wrongly identified as a significant feature. Hence, α is set to 7.879. Besides, only the features that appear in more than one-tenth of the documents in the corresponding period would calculate their χ^2 value. This is because rare features are difficult to estimate correctly and this can reduce significant computational cost. Algorithm 3 outlines the procedure of selecting positive training articles.

7.3 Selecting Negative Training Examples

For all of the machine learning problems, it is necessary to present both the positive and negative training data to the system, or else nothing could be learned. In this section, a heuristic for selecting negative training examples is presented.

Define X be the trend which is either rise (R) or drop (D), i.e. $X \in \{R, D\}$. Let $\mathcal{D}_{X,+}$ and $\mathcal{D}_{X,-}$ denote the sets containing the positive training examples and the negative training examples in X, respectively, i.e. $\mathcal{D}_{X,+} \in \{\mathcal{D}_{R,+}, \mathcal{D}_{D,+}\}$ and $\mathcal{D}_{X,-} \in \{\mathcal{D}_{R,-}, \mathcal{D}_{D,-}\}$.

In the previous section, we have introduced an selection algorithm (Algorithm 3) for generating two sets of supportive documents, $\mathcal{D}_{R,+}$ and $\mathcal{D}_{D,+}$, which serve as positive training examples. However, we do not have any negative training examples. One cannot simply assume that documents which are not in $\mathcal{D}_{X,+}$ belong to $\mathcal{D}_{X,-}$.

Algorithm 3 genPositive($\mathcal{D}$, $\mathcal{T}'_m$) – Generate positive training examples from a collection of documents $\mathcal{D}$ given a segmented time series $\mathcal{T}'_m$.

```
for each S_k in T'_m do
    L_{feature,k} = φ;
    D_{S_k} = φ;
    if t(d_i) ≥ t_b(S_k) and t(d_i) < t_e(S_k) then
        Assign d_i to D_{S_k};
    end if
    for each f_i ∈ D_{S_k} do
        if χ²(f_i) ≥ α then
            append f_i to L_{feature,k};
        end if
    end for
end for
D_{R,+} = φ;
D_{D,+} = φ;
for each f_j ∈ L_{feature,k} do
    if f_j ∈ {d_i | d_i ∈ D_{S_k}} then
        if slope(S_k) > 0 then
            Assign d_i to D_{R,+};
        else
            Assign d_i to D_{D,+};
        end if
    end if
end for
```

Algorithm 4 outlines the whole procedure for selecting $\mathcal{D}_{X,-}$. Let $|\mathcal{D}_{X,+}|$ and $|D|$ be the number of documents in $\mathcal{D}_{X,+}$ and $\mathcal{D}$, respectively; $n_{X,+}(f_j)$ and $n(f_j)$ be the number of documents containing f_j in $\mathcal{D}_{X,+}$ and $\mathcal{D}$, respectively. Define *positive feature* be the feature that are believed to play an active role in $\mathcal{D}_{X,+}$. Intuitively, the negative training examples should not contain any of the positive features. Hence, the first step of selecting negative training examples is to figure out these positive features.

To achieve our goal, a weight, $\hat{\omega}(f_j)$, which associated with the feature f_j

Algorithm 4 genNegative($\mathcal{D}$, $\mathcal{D}_{X,+}$) – Generate negative training examples from a collection of documents $\mathcal{D}$ given a set of positive training examples.

1: $\mathcal{L}_{feature,X} = \phi$;
2: **for each** $f_j \in \mathcal{D}_{X,+}$ **do**
3: $\quad \hat{\omega}(f_j) = \big(\omega_X(f_j)/\omega(f_j)\big) \cdot \big(\omega_X(f_j) - \omega(f_j)\big)$;
4: **end for**
5: $\gamma \ni \max_{f_j \in d_i}\{\hat{\omega}(f_j)\}, \forall d_i \in \mathcal{D}_{X,+}$;
6: $\theta = \min_{f_j \in \gamma}\{\hat{\omega}(f_j)\}$;
7: **for each** $f_j \in \mathcal{D}_{X,+}$ **do**
8: $\quad$ **if** $\hat{\omega}(f_j) < \theta$ **then**
9: $\quad\quad$ Append f_j to $\mathcal{L}_{feature,X}$;
10: $\quad$ **end if**
11: **end for**
12: $\mathcal{D}_{X,-} = \phi$;
13: **for each** $f_j \in d_i$ **do**
14: $\quad$ **if** $f_j \cap \mathcal{L}_{feature,X} = \phi$ **then**
15: $\quad\quad$ Assign d_i to $\mathcal{D}_{X,-}$;
16: $\quad$ **end if**
17: **end for**

in $\mathcal{D}_{X,+}$ is computed, so as to account for its relative importance in that set. A higher weight indicates that the feature is more representative in there. This weight is computed by the following schemes:

$$\hat{\omega}(f_j) = \frac{\omega_{X,+}(f_j)}{\omega(f_j)} \cdot (\omega_{X,+}(f_j) - \omega(f_j)) \tag{7.2}$$

$$\omega_{X,+}(f_j) = \frac{n_{X,+}(f_j)}{|\mathcal{D}_{X,+}|} \tag{7.3}$$

$$\omega(f_j) = \frac{nf_j}{|D|} \tag{7.4}$$

Some famous techniques for rating feature importance, such as entropy, information gain and mutual information, cannot be applied here. This is because all of these techniques require to estimate the expected probability distribution

between the positive training documents and the negative training documents, which is impossible to know at this stage.

After the weights of the features are generated, a threshold cut is applied on these weighted features, such that only the features with $\hat{\omega}(f_j) > \theta$ would be considered as positive features. Let $\mathcal{L}_{feature,X}$ be a list that stores all of the positive features in X. Here, θ is computed as:

$$\theta = \min_{f_j \in \gamma} \left\{ \hat{\omega}(f_j) \right\} \tag{7.5}$$

$$\gamma \ni \max_{f_j \in d_i} \left\{ \hat{\omega}(f_j) \right\}, \forall d_i \in \mathcal{D}_{X,+} \tag{7.6}$$

In other words, θ is base on minimax principle such that it is the lowest value among the highest weights of the features within a document in $\mathcal{D}_{X,+}$. This heuristic could reasonably approximate negative training examples which does not significantly affect the classification performance. Finally, $\mathcal{D}_{X,-}$ is obtained by including the documents which do not have any the features listed in $\mathcal{L}_{feature,X}$, i.e. $d_i \in \{\mathcal{D}_{X,-} \mid f_j \cap \mathcal{L}_{feature,X} = \phi, \forall f_j \in d_i\}$

Chapter 8

System Learning

When we received a piece of news article, we have to determine it will trigger what kind of movement on the stock time series. Hence, we can formulate this problem as a classification problem, where we are trying to classify the impact of the news article. For the existing classification approaches, Naive Bayes is properly the best in term of computational efficiency. However, its quality is very low. At the mean time, for shortcoming of it is the very expensive training cost. In this chapter, a new classification approach, called *Discriminative Category Matching* (DCM), is proposed. DCM is a similarity based text classifier which is as efficiency as NB, meanwhile its quality is comparable to that of SVM.

Note that the presentation in the following sections deliberately keep the description of DCM at a high level text classifier, such that the heuristic proposed in this chapter is not just suitable to be used within this dissertation, but also suitable for solving any text classification problem. Hence, instead of viewing DCM as a component in a system, it should be better regarded as a general purpose text classifier.

Symbol	Meaning
c_k	A category
d_i	A document
$\tilde{c}_k$	A category sketch
$\tilde{d}_i$	A document sketch
f_j	A Feature
n_{c_k}	The number of documents in c_k
l_{d_i}	The total number of features in d_i
N	The total number of different categories
tf_{f_j,d_i}	The number of different features in d_i
df_{f_j,c_k}	The number of documents containing f_j in c_k

Table 8.1: List of Symbols and their meanings that would be used throughout this chapter.

8.1 Similarity Based Classification Approach

DCM is a similarity based classifier which tries to classifier an unseen document to some pre-defined categories according the degree of similarity among them. Table 8.1 shows a list of symbols and their meanings that would be used throughout this chapter.

Let d_i and c_k be a document and a category, respectively; $\tilde{d}_i$ and $\tilde{c}_k$ be the corresponding document sketch and category sketch, respectively. A sketch is an approximator which stores the information about the characteristics of the corresponding document or category. A high similarity between $\tilde{c}_k$ and $\tilde{d}_i$ indicates that d_i should be classified to c_k. To improve the performance, the information that store in a sketch must be minimized. Thus, in DCM the information store in a category sketch $\tilde{c}_k$ only includes the features appearing in the documents archived. The category sketch, c_k, and document sketch, d_i, are in the following

form:

$$\tilde{d}_i = \langle (f_1, w_1), (f_2, w_2), \ldots, (f_n, w_n) \rangle \tag{8.1}$$

$$\tilde{c}_k = \langle (f_1, W_1), (f_2, W_2), \ldots, (f_n, W_n) \rangle \tag{8.2}$$

Given $\tilde{d}_i$ and $\tilde{c}_k$, their similarity are measured by Jaccard coefficient:

$$S(\tilde{d}_i, \tilde{c}_k) = \frac{\sum_{w_j \neq 0, W_j \in c_k} (w_j \cdot W_j)}{\sum_{w_j \neq 0} w_j^2 + \sum_{w_j \neq 0, W_j \in c_k} W_j^2 - \sum_{w_j \neq 0, W_j \in c_k} (w_j \cdot W_j)} \tag{8.3}$$

Jaccard coefficient is chosen because it expresses the degree of overlapping between $\tilde{d}_i$ and $\tilde{c}_k$ as the proportion of overlapping between them. It provides both intuitive and practical fitness to our model.

Given N predefined categories, $c_1, c_2, \cdots, c_N$, DCM first computes $\tilde{d}_i$ on d_i and sorts all category sketches in descending order by following Equation (8.3). Let $\mathcal{L}_{category} = \tilde{c}'_1, \tilde{c}'_2, \ldots, \tilde{c}'_N$ be such a list. Finally, DCM determines the top K category sketches $\mathcal{L}_{category}$, $\tilde{c}'_1, \tilde{c}'_2, \ldots, \tilde{c}'_K$, that are most similar with the unseen document sketch, $\tilde{d}_i$, by following the mean square error, $E(\tilde{c}_k)$:

$$E(\tilde{c}_k) = \frac{1}{2}(S(\tilde{d}_i, \tilde{c}_k)^2 - S(\tilde{d}_i, \tilde{c}_{k+1})^2) \tag{8.4}$$

and assign d_i to $\tilde{c}'_1, \tilde{c}'_2, \ldots, \tilde{c}'_K$, such that $E(\tilde{c}'_x) \leq E(\tilde{c}'_{x+1})$ for $x < K$ and $E(\tilde{c}'_K) > E(\tilde{c}'_{K+1})$. In the following sections, I will discuss how the category sketch $\tilde{c}_k$ and document sketch $\tilde{d}_i$ are generated.

8.2 Category Sketch Generation

The weight, W_j, for feature f_j in category c_k is computed as follows:

$$W_j = AI(f_j, c_k) \cdot \left(\sqrt{2} \cdot \frac{WC(f_j, c_k)^2 \cdot CC(f_j)^2}{\sqrt{WC(f_j, c_k)^2 + CC(f_j)^2}} \right) \tag{8.5}$$

Here, $\sqrt{2}$ is used for normalization such that $0 \leq W_j \leq 1$. $WC(f_j, c_k)$, $CC(f_j)$ and $AI(f_j, c_k)$ are known as *Within-Category Coefficient*, *Cross-Category Coefficient* and *Average-Importance Coefficient*, respectively. In the following, I would discuss their physical meanings as well as how to compute them.

8.2.1 Within-Category Coefficient

The Within-Category Coefficient, $WC(f_j, c_k)$, is used to measure the relative importance of feature f_j in category c_k:

$$WC(f_j, c_k) = \frac{\log_2(df_{f_j, c_k} + 1)}{\log_2(n_{c_k} + 1)} \tag{8.6}$$

$WC(f_j, c_k)$ reflects the fact that features appearing frequently within a category is critical in term of classification. In Equation (8.6), both the numerator and the denominator are logarithmic. This is because the frequency of a feature appearing over many documents is rare. Similar finding is also reported in [24].

8.2.2 Cross-Category Coefficient

The Cross-category Coefficient, $CC(f_j)$, is used to measure the relative importance of feature f_j among all categories:

$$CC(f_j) = \frac{1}{\log_2 N} \cdot \log_2 \frac{N}{\delta} \tag{8.7}$$

$$\delta = \frac{\sum_{k=1}^{N} WC(f_j, c_k)}{\max_{k \in [1..N]}\{WC(f_j, c_k)\}} \tag{8.8}$$

where the term $1/\log_2(N)$ in Equation (8.7) is used for normalization such that $0 \leq CC(f_j) \leq 1$. Note that in Equation (8.8), the summation, $\sum$, gathers the total importance of a feature across all categories, and the maximum, max, averages the summation value.

The significant of $CC(f_j)$ is that it gives a global view to weight the importance of a feature across all categories. In fact, if a feature is regarded as important in many categories, then this feature is obviously not important for classification. For example, suppose that there exist two features, f_a and f_b, where f_a appears in m_1 categories and f_b appears in m_2 categories, two possibilities exist:

- **Case-1** ($m_1 \ll m_2$): Obviously, f_a provides far more precious information for text classification than f_b, as f_a appears only in a limited number of categories. In other words, a feature is more valuable if its occurrence is skewed.

- **Case-2** ($m_1 = m_2$): In this case, it would not be realistic to weight the importance of a feature by simply counting the number of categories that it appears, as proposed by Yamamoto [63]. Intuitively, a higher weight

should be assigned to the feature where more documents in the category(s) contains it. Equation (8.8) just capture this idea.

8.2.3 Average-Importance Coefficient

The Average-Importance Coefficient, $AI(f_j, c_k)$, is used to measure the average importance of a feature, f_j, among all individual documents in a category, c_k:

$$AI(f_j, c_k) = \left(\frac{\sum_{d_i \in c_k} \omega_{f_j,d_i}}{df_{f_j,c_k}} \right)^{\beta_{f_j,c_k}} \tag{8.9}$$

$$\omega_{f_j,d_i} = \frac{\log_2(tf_{f_j,d_i} + 1)}{\log_2(l_{d_i} + 1)} \tag{8.10}$$

Note that a feature appears n times does not imply that it is n times more important [22, 48], thus a logarithmic relationship is taken rather than a linear relationship. In Equation (8.9), the term within the bracket averages the weights of the feature, f_j, among all documents in the category c_k. The unbias estimator, β_{f_j,c_k}, determines the suitability of this average:

$$\beta_{f_j,c_k} = \frac{1}{1 + WC(f_j, c_k)} \tag{8.11}$$

For instance, given two features: f_a and f_b, such that f_a appears in most of the documents in the category c_k but f_b only appears in a few documents there. Obviously, it is more confident to declare that the average weight of f_a is more likely to reflect the true status (importance) of it than that of f_b, as the sample size is larger.

Unlike $WC(f_j, c_k)$ and $CC(f_j)$, which are designed at category level, here $AI(f_j, c_k)$ handles the importance of a feature within an individual document. It

reduces the discrepancy among weights, and increase the recall and precision of the classifier. As for memory consumption, DCM only needs to keep df_{f_j,c_k} and ω_{f_j,c_k} in memory.

8.3 Document Sketch Generation

In text classification, we need to figure out the representative of a feature in the training set, as well as the new incoming document which is waiting for being classified. In the previous section, I have proposed a technique for handling the former case. For the latest situation, a standard procedure for solving it is by considering the features' distribution of the incoming document. However, in practice, we usually deal with this issue in a document basis (one document at a time). Thus, the sample size is insufficient. Document sketch is proposed to estimate the truth importance of the features in an incoming document.

The most simplest and efficient way to estimate the weight, w_j, of a feature, f_j, appears in an unseen document, d_i, is to average the weights of the corresponding feature f_j, over all of the training document in all categories. Let $\overline{AI}(f_j)$ be this averaged weight:

$$\overline{AI}(f_j) = \left(\frac{\sum_{k=1}^{N} \sum_{W_j \in c_k} W_j}{\sum_{k=1}^{N} df_{f_j,c_k}} \right)^{\overline{\beta}_{f_j}} \tag{8.12}$$

where $\overline{\beta}_{f_j}$ is an unbias estimator which is introduced in a similar fashion as

Equation (8.11):

$$\overline{\beta}_{f_j} = \frac{1}{1 + \overline{WC}(f_j)} \tag{8.13}$$

$$\overline{WC}(f_j) = \frac{\log_2(\sum_{k=1}^{N} df_{f_j,c_k} + 1)}{\log_2(\sum_{k=1}^{N} n_{c_k} + 1)} \tag{8.14}$$

However, only the above is insufficient. We cannot assume that every incoming document that contains the feature f_j shares the same distribution. In other words, there are some risks in using Equation (8.12). Thus, a geometric distribution is formulated [38]:

$$R_{f_j} = \left(\frac{1}{1 + \overline{tf}_{f_j}}\right) \times \left(\frac{\overline{tf}_{f_j}}{1 + \overline{tf}_{f_j}}\right)^{tf_{f_j,d_i}} \tag{8.15}$$

$$\overline{tf}_{f_j} = \overline{AI}(f_j) \times l_{d_i} \tag{8.16}$$

where $\overline{tf}_{f_j}$ is the estimated average value for the number of times the feature f_j appears in a randomly picked document from the training data.

Finally, the weight, w_j, of feature f_j in $\tilde{d}_i$ is computed by combining Equation (8.10) and Equation (8.12) to Equation (8.16):

$$w_j = \omega_{f_j,d_i}^{1-R_{f_j}} \times \overline{AI}_{f_j}^{R_{f_j}} \tag{8.17}$$

Thus, $\tilde{d}_i$ is generated. The similarity between $\tilde{d}_i$ and $\tilde{c}_k$ could then be measured by using Equation (8.3). The whole process of text classification is achieved.

Chapter 9

System Operation

Based on the techniques discussed in the previous three chapters, we now pack all of the stuff together and process to the final step – system operation. This chapter discuss how the system operates.

9.1 System Operation

Recall that in Chapter 7, after the text pre-processing, four sets of documents are resulted: 1) Documents that support the rise movements ($\mathcal{D}_{R,+}$); 2) Documents that do nothing with the rise movement ($\mathcal{D}_{R,-}$); 3) Documents that support the drop movements ($\mathcal{D}_{D,+}$); and 4) Documents that do nothing with the drop movements ($\mathcal{D}_{D,-}$). Here, DCM, which is proposed in Chapter 8, is used for learning the relationships among these four sets of documents. Four categories are defined: 1) Category $c_{R,+}$ that contains $\mathcal{D}_{R,+}$; 2) Category $c_{R,-}$ that contains $\mathcal{D}_{R,-}$; 3) Category $c_{D,+}$ that contains $\mathcal{D}_{D,+}$; and 4) Category $c_{D,-}$ that contains $\mathcal{D}_{R,-}$.

For any previously unseen news articles, $\tilde{d}'_i$, it would be passed to DCM for determining which of the category(s) should it belong. By doing so, we could predict what kinds of impacts that this unseen article would trigger. Here, two possibilities of the classification result may obtain:

1. $\tilde{d}'_i$ **belongs to only one of the category** In this situation, the impact of $\tilde{d}'_i$ is as same as the type of the category. For instance, if $\tilde{d}'_i$ belongs to $c_{R,+}$ (or $c_{D,+}$), we conclude that $\tilde{d}'_i$ will trigger a rise (or drop) movement on the stock time series. However, if $\tilde{d}'_i$ belongs to either $c_{R,-}$ or $c_{D,-}$, we conclude that it does not have any impacts.

2. $\tilde{d}'_i$ **belongs to multiple categories** In this situation, we further investigate if some the categories that $\tilde{d}'_i$ is assigned to are *conflict*. The full list of categories that are considered as conflict are: 1) $\{c_{R,+}$ and $c_{D,+}\}$; 2) $\{c_{R,+}$ and $c_{R,-}\}$; and 3) $\{c_{D,+}$ and $c_{D,-}\}$. If so, the impact of $\tilde{d}'_i$ is ambiguous, and that piece of news article is ignored. If not, the impact of $\tilde{d}'_i$ belongs to the type of the positive category that it has been assigned to (i.e. either belongs to $c_{R,+}$ or $c_{D,+}$).

Part IV

Results and Discussions

Before you criticize someone, you should walk a mile in their shoes. That way, when you criticize them, you are a mile away from them, and you have their shoes

Frieda Norris

Chapter 10

Evaluations

A prototype system using Java™ is developed to evaluate the proposed approach described in Part III. All of the experiments are conducted on a Sun Blade-1000 workstation running Solaris 2.8 with 512MB physical memory and with a 750MHz Ultra-SPARC-III CPU. Intra-day stock prices and real-time news articles are archived through Reuters Market 3000 Extra[1] from 20^{th} January 2003 to 20^{th} June 2003. All data are stored into IBM DB2 Version 7.1[2].

For the real-time news articles, there are more than 350,000 documents archived. Note that Reuters has assigned to which sectors, countries, etc, the news articles should belong. Therefore, we do not need to worry about how these news articles should be organized. All features from the news articles are stemmed and converted to lower cases, in which punctuation and stop-words are removed, numbers, web page addresses and email addresses are ignored.

For the stock data, the intra-day price of all the Hong Kong stocks are

[1] `http://www.reuters.com`
[2] `http://www.ibm.com`

recorded[3],[4]. According to the observation given by technical analysis that price movements associated with light volumes denotes only temporal movements, but not trends[5], thus, for each stock, the transactions that are associated with light volumes are ignored.

10.1 Time Series Evaluations

Figure 10.1 shows the typical results of applying the t-test based split and merge segmentation algorithm. Due to the space limit, only the results of three representative stocks are shown. They are: 1) Cheung Kong (0001.HK); 2) Cathay Pacific (0293.HK) and 3) TVB (0511.HK). The unmodified stock data are shown on the left while the segmented data are shown on the right. This kind of representation of the time series allows us to make decisions that are based on variable length intervals lasting for days to weeks, rather that minute-by-minute or even second-by-second. One could see that the trends generated are quite reasonable and suitable. Note that the longest trend last for 2 weeks while the shortest one last for 3 days. This could assure that all of the trends are tertiary movements.

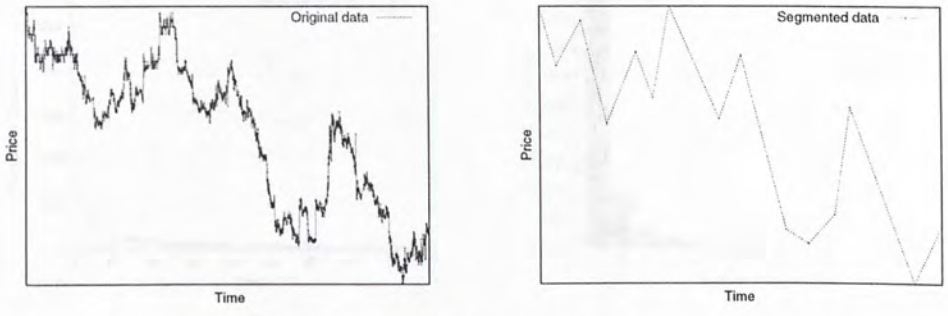
10.2 Classifier Evaluations

Instead of using my own news article collection for evaluating the performance of DCM, two benchmarks, Reuters-21578 and Newsgroup-20, are also being used. Table 10.1 summarized these data sets. Their details are as follows:

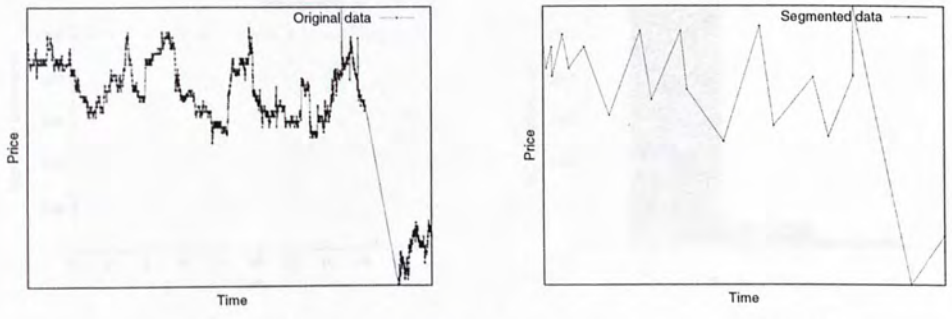
[3]The stocks which have too few transaction records are ignored. This is simply because there are not enough data for training and/or evaluation.

[4]Each stock belongs to one of the categories listed in Appendix A

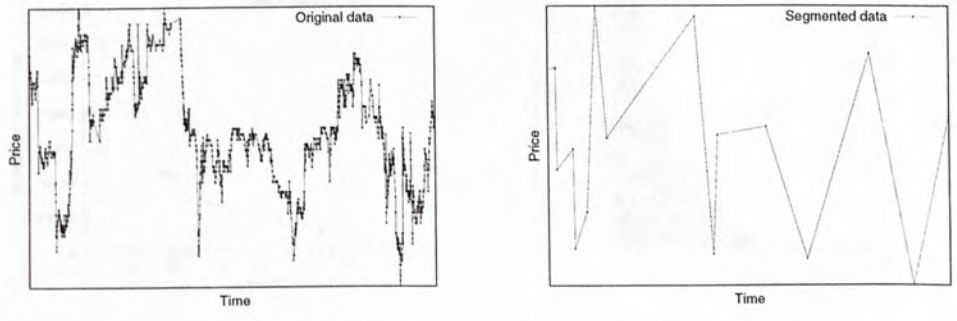
[5]Please refer to Chapter 4 Section 1 for details.



(a) Before and after segmentation for Cheung Kong (0001.HK).

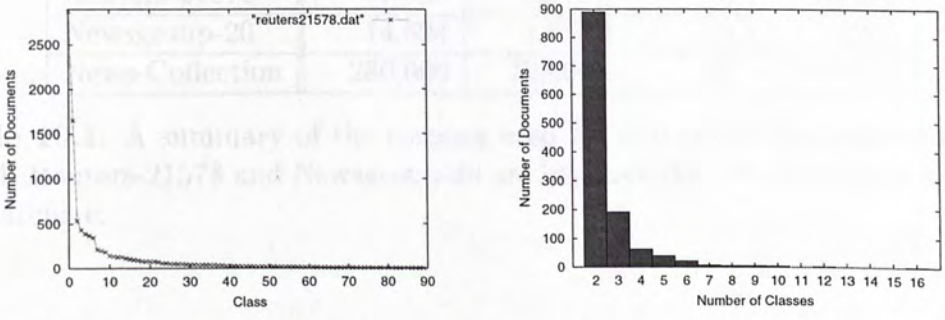


(b) Before and after segmentation for Cathay Pacific (0293.HK).

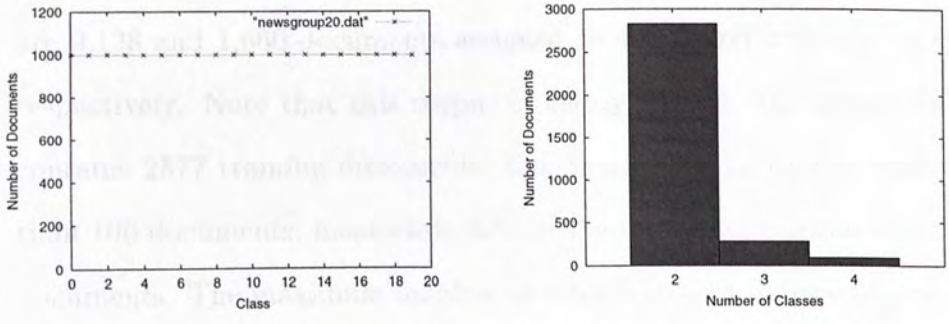


(c) Before and after segmentation for TVB (0511.HK).

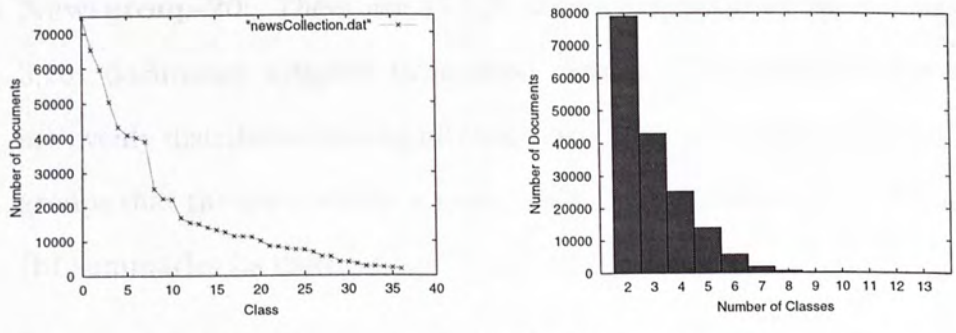
Figure 10.1: Before and after applying the t-test based split and merge segmentation algorithm. On the left: The original time series. On the right: The segmented time series. Three stocks are selected to report here: a) Cheung Kong (0001.HK); (b) Cathay Pacific (0293.HK); and (c) TVB (0511.HK).



(a) The document distribution of Reuters-21578.



(b) The document distribution of Newsgroup-20.



(c) The document distribution of News-Collection.

Figure 10.2: Document distribution. On the left: the number of documents assigned to each category. The X-axis (category) is ranked from common to rare. On the right: the number of documents assigned to multiple categories. (a) The distribution of Reuters-21578. (b) The distribution of Newsgroup-20. (c) The distribution of News-Collection.

Dataset	Training	Testing	Classes	Features
Reuters-21578	7,769	3,019	90	13,270
Newsgroup-20	14,694	1,633	20	25,245
News-Collection	280,000	70,000	38	206,897

Table 10.1: A summary of the corpora used for evaluating the performance of DCM. Reuters-21578 and Newsgroup-20 are benchmarks. News-Collection is our own archive.

- **Reuters-21578**: The ModApte-split version is used and the categories that have at least one document for training and evaluation are selected. There are 9,128 and 1,660 documents assigned to single and multiple categories, respectively. Note that this corpus is highly skewed: the largest category contains 2877 training documents, but 85% of the categories contain less than 100 documents, meanwhile 33% of the categories contain less than 10 documents. The maximum number of categories that a piece of new article is assigned to is 16. Figure 10.2 (a) summaries its distribution.

- **Newsgroup-20**: There are 13,126 documents assigned to one class and 3,201 documents assigned to multiple classes. The number of documents are evenly distributed among all categories. The maximum number of newsgroups that the same article is being posted simultaneously is 4. Figure 10.2 (b) summaries its distribution.

- **News-Collection**:

  A very large set of news articles that are archived through Reuters directly is discussed in the beginning of this chapter. Our task is to assign the archived news articles to at least one of the 38 Morgan Stanley Capital

International (MSCI) code[6]. Note that Reuters has already assigned the suitable MSCI code(s) to the news articles. Figure 10.2 (c) summaries its distribution.

The performance of DCM is compared with two other popular classification approaches: Naive Bayes (NB) and support vectors machine (SVM). NB is chosen because it performs very good in terms of efficiency. SVM is chosen because it is the best reported text classification algorithm in terms of quality up-to-date. The setting of NB, SVM and DCM are as follows:

- **Naive Bayes** The Multinomial Mixture model is used for training and evaluation [35, 64]. Features are selected based on information gain (IG). Recall that classifier building/updating requires re-selecting the suitable features by re-calculating the IG from scratch. This consumes most of the computational cost.

- **Support Vectors Machine** The classifier is trained and evaluated by using the SVM^{light} package [26]. The weight of each feature is calculated by standard $tf \cdot idf$ scheme and are normalized into unit length [48]. The classifier is updated based on the newly received documents and the previously learned support vectors identical to the approach described by Syed et al. [52]. The classifier building/updating requires re-calculating the feature weights from scratch. This leads to a significant overhead cost.

- **Discriminative Category Matching** It is implemented using Java^{TM} with 300 lines implementation. Neither feature selection nor weighing is

[6]Please refer to Appendix B for the details of MSCI code

necessary. No any other extra precessing is necessary in classifier building/updating.

Following the existing evaluation approaches, for measuring the quality of the classifier, micro-recall (m-R), macro-recall (M-R), micro-precision (m-p) and macro-precision (M-P) are used. Marco-value gives an equal weight to the performance on every category, regardless of how common or rare a category is. On the other hand, micro-value gives an equal weight to the performance of every document, thus favoring the performance of the large category [51, 64]. In order to have a harmonic average over them, F1 measure is taken [51]. [7]

10.2.1 Batch Classification Evaluation

In this evaluation, the training set and evaluation set of the data are divided as follows:

- **Reuters-21578** Since the ModApte-split version of this data set has already maintained some rules about how to separate the documents into training set and evaluation set, I adopted these rules directly.
- **Newsgroup-20** The first 70% of the data are used for training while the remaining 30% are used for evaluation.
- **News-Collection** Similar to Newsgroup-20, the first 70% of the data are used for training while the remaining 30% are used for evaluation.

Table 10.2 summaries the results of NB, SVM and DCM. As expected, SVM outperforms NB significantly in all experiments. This replicates the previous

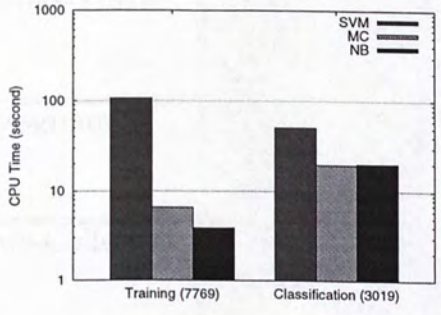
[7] Please refer to Appendix C for issues related to precision, recall and F1.

	Method	m-P	m-R	m-F1	M-P	M-R	M-F1
Reuters-21578	SVM	**0.912**	0.802	**0.857**	**0.533**	0.479	0.505
	DCM	0.807	**0.851**	0.828	0.498	**0.650**	**0.564**
	NB	0.776	0.751	0.763	0.401	0.378	0.381
Newsgroup-20	SVM	**0.882**	0.617	**0.726**	**0.873**	0.618	**0.724**
	DCM	0.721	**0.669**	0.694	0.722	**0.686**	0.703
	NB	0.671	0.608	0.638	0.670	0.639	0.655
News-Collection	SVM	**0.658**	0.632	**0.643**	**0.518**	0.495	**0.503**
	DCM	0.540	**0.671**	0.591	0.406	**0.579**	0.487
	NB	0.603	0.489	0.536	0.583	0.335	0.425

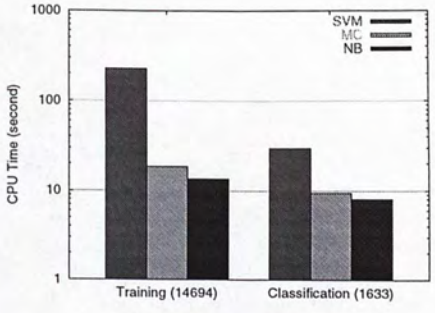
Table 10.2: Results of the classification effectiveness in the batch classification.

research findings [26, 36, 64]. Comparing with SVM, DCM performs superior in Reuters-21578. For Newsgroup-20, DCM performs better at macro-level, whereas SVM performs better at micro-level. As macro-level takes a global view across categories, it suggests that DCM may perform consistent over all categories with different sizes. However, a direct comparison between micro-level values and macro-level values is inappropriate since they focus on different aspects. Since neither SVM nor DCM dominates in both measurements, we cannot conclude that which approach performs better in Newsgroup-20. It all depends on which issue one may concern with.

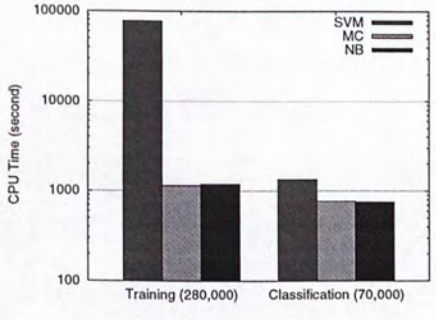
Figure 10.3 shows the CPU cost (including both classifier building and preprocessing costs) of the three approaches. Note that DCM and NB outperform SVM significantly in both training and operation. Furthermore, this observation becomes far more clear as the size of the data set increased.



(a) Reuters-21578



(b) Newsgroup-20



(c) News-Collection

Figure 10.3: Results of the computational efficiency of the batch classification.

10.2.2 Online Classification Evaluation

This experiment mimic an online situation such that the classifiers require to update frequently when they have been operated for an arbitrary period. Documents are divided into 10 equal-sized batches. A Classifier is built by using the first batch, and evaluate it using the second batch. Then, we update the classifier using the second batch, and evaluate it using the third batch, and so on so forth. Note that the CPU cost reported in this dissertation includes the classifier

	Method	**m-P**	**m-R**	**m-F1**	**M-P**	**M-R**	**M-F1**
Reuters-21578	SVM	**0.840**	0.789	**0.824**	0.543	0.609	0.569
	DCM	0.800	**0.831**	0.815	**0.557**	**0.721**	**0.628**
	NB	0.741	0.799	0.767	0.336	0.524	0.406
Newsgroup-28	SVM	**0.722**	0.625	**0.685**	**0.694**	**0.619**	0.667
	DCM	0.693	**0.643**	0.667	**0.694**	0.609	**0.678**
	NB	0.632	0.585	0.597	0.633	0.623	0.620
News-Collection	SVM	**0.633**	0.582	**0.604**	**0.526**	0.478	**0.502**
	DCM	0.539	**0.668**	0.587	0.485	**0.502**	0.497
	NB	0.603	0.489	0.556	0.493	0.367	0.378

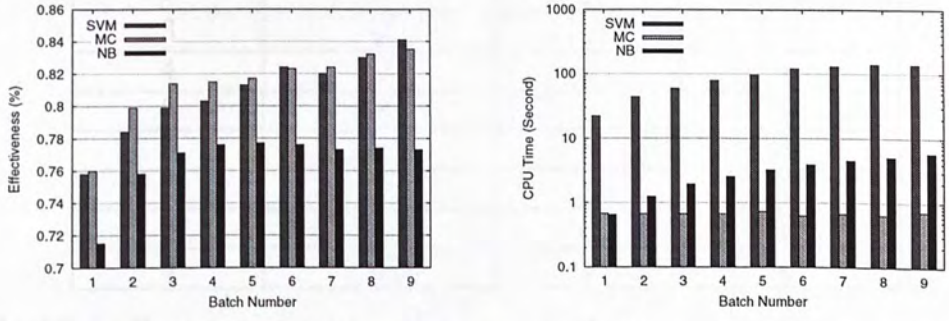
Table 10.3: Results of the classification effectiveness in the online classification.

building/updating and operation costs.

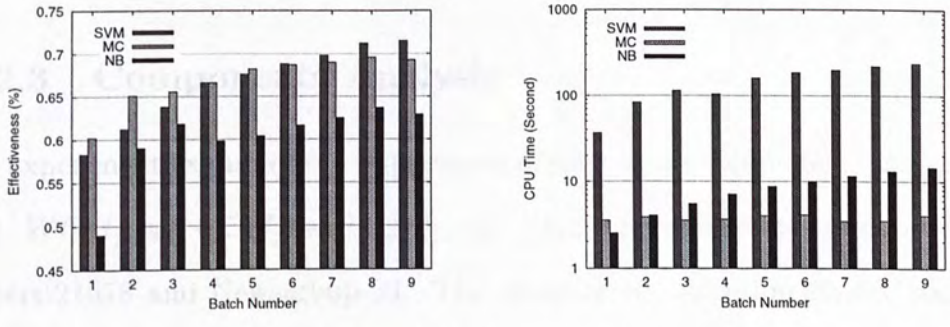
Table 10.3 shows the average accuracy for each of the three classification approaches. The accuracy of DCM is similar to that of SVM, and is far superior than NB. In particular, DCM performs significantly well in terms of macro-recall (M-R) and micro-Recall (m-R). Although DCM does not outperform SVM in terms of accuracy, its low CPU cost in both the classifier updating and operation makes it highly recommended for online text classification (Figure 10.4).

Figure 10.4 (a), (c) and (e) show the m-F1 of the three algorithms whenever a new batch arrives. For all of the algorithms, m-F1 increases from the first batch and become saturated after 4-5 batches. This is because the classifiers obtain a relatively sufficient examples from the corresponding corpus. In all cases, NB always performs inferior than the other two approaches. The accuracy of DCM and SVM are similar.

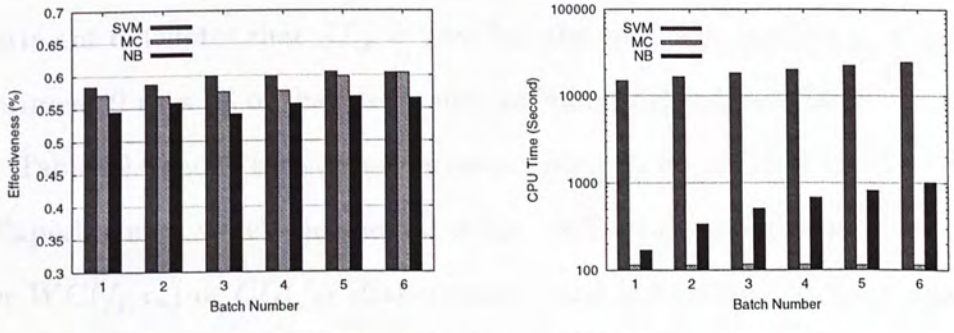
Figure 10.4 (b), (d) and (f) show the CPU cost (including the classifier building/updating and operation costs) of the three approaches. Note that DCM outperforms both NB and SVM significantly.



(a) Results for Reuters-21578.



(b) Results for Newsgroup-20.



(c) Results for News-Collection.

Figure 10.4: Results of the F1 score and computational efficiency of the online classification. On the left: The classification accuracy based on F1 measurement. On the right: The classification efficiency measured in CPU time (in second).

Experiment	use $WC(f_j, c_k)$	use $CC(f_j)$	use $AI(f_j, c_k)$
1	√	×	×
2	×	√	×
3	√	√	×
4	√	√	+
5	×	×	√
6	×	×	+
7	√	×	√
8	×	√	√
9	√	√	√

Table 10.4: Experiment settings to examine the usefulness of the components $WC(f_j, c_k)$, $CC(f_j)$, $AI(f_j, c_k)$ and β_{f_j,c_k}.

10.2.3 Components Analysis

This experiment examines the importance of each of the components in Equation (8.3): $WC(f_j, c_k)$, $CC(f_j)$ and $AI(f_j, c_k)$. Nine experiments are conducted using Reuters-21578 and Newsgroup-20. The setup of the experiments are shown in Table 10.2.3. Here, a tick denotes that the corresponding component is present, whereas a cross denotes for an absent. The "+" sign in Expereiment 4 and Experiment 6 denotes that $AI_{i,k}$ is used but the risk estimator, β_{f_j,c_k}, is omitted. Experiment 9 uses all of the coefficients including the risk estimator.

Table 10.6 shows the evaluation results using Reuters-21578 and Newsgroup-20. Experiment 9, which includes all of the coefficients, performs the best. Using either $WC(f_j, c_k)$ or $CC(f_j)$ (Experiment 1 and Experiment 2) gives unacceptable results. Interestingly, a combination of them yields a significant improvement (Experiment 3). Ignoring the average estimator (Experiment 6) results an inferior result. In conclude, all of the components have their own contributions, and none of them can be ignored.

Experiment	Reuters-21578		Newsgroup-20	
	m-F1	M-F1	m-F1	M-F1
1	0.007	0.391	0.005	0.234
2	0.028	0.200	0.018	0.125
3	0.745	0.350	0.514	0.459
4	0.815	0.515	0.604	0.640
5	0.780	0.375	0.584	0.578
6	0.699	0.318	0.510	0.513
7	0.775	0.374	0.531	0.578
8	0.686	0.350	0.482	0.508
9	**0.828**	**0.564**	**0.694**	**0.703**

Table 10.5: Results of the evaluation for the usefulness of the components $WC(f_j, c_k)$, $CC(f_j)$, $AI(f_j, c_k)$ and β_{f_j,c_k}.

10.2.4 Document Sketch Analysis

In DCM, a document sketch is influenced by the entire feature distribution in the training data. The unbias estimator for unseen document, $\overline{\beta}_{f_j}$, in Equation (8.12) plays an important role in formulating the weights of the features in a document sketch. The experiments are conducted to evaluate the necessity and the influence of $\overline{\beta}_{f_j}$: Experiment 1 denotes that $\overline{\beta}_{f_j}$ is used, whereas Experiment 2 denotes that is $\overline{\beta}_{f_j}$ ignored (i.e. $\overline{\beta}_{f_j} = 1$).

Table 10.6 shows the evaluation results. As expected, The performance of Experiment 1 far outperforms Experiment 2. It shows that $\overline{\beta}_{f_j}$ is necessary and significant in text classification.

10.3 Prediction Evaluations

One of the best way to evaluate the reliability of a prediction system is to conduct a market simulation which mimics the behaviors of stock traders using real-life

Experiment	Reuters-21578		Newsgroup-20		News-Collection	
	m-F1	M-F1	m-F1	M-F1	m-F1	M-F1
1	**0.828**	**0.564**	bf 0.694	**0.703**	**0.591**	**0.487**
2	0.803	0.552	0.673	0.682	0.562	0.470

Table 10.6: The necessity of $\overline{\beta}_{f_j}$ in the document sketch.

data. As a result, two market simulations[8] are conducted:

- **Simulation 1: Proposed System**: Shares are bought or sold based solely on the content of the news articles. Two strategies are adopted:
  - For each stock, if the prediction of its upcoming trend is positive, then shares of it are bought immediately. The shares would be sold after holding for m day(s).
  - For each stock, if the prediction of its upcoming trend is negative, then shares of that stock are sold for short. The shares would be bought back after m day(s).

  A detailed analysis of how m affects the evaluation results is given in Section 10.3.2. In this section, m is set to 3 working days for simplicity. If the market is closed when the decision is made, then shares will be bought or sold in the beginning of the next active trading day.

- **Simulation 2: Buy-and-Hold Test**: For each stock, shares of that stock is bought at the beginning of the evaluation period. At the end of the evaluation period, all of the shares remain on hand are sold. This simulation serves as a base-line comparison which is used to demonstrate the do-nothing strategy.

[8]The assumption of zero transaction is carried out.

	Simulation 1	**Simulation 2**
Accumulative r	**28.06**	-20.56
Stand. Dev. of r	**2.40**	1.15
Maximum r	**10.42**	2.21
Minimum r	**-4.83**	-23.10
Top ten average r	**8.18**	1.11
Least ten average r	**-3.69**	-18.56

Table 10.7: The overall evaluation results of the two market simulation. Here, r is the rate of return.

In the above market simulations, rate of return, r, is calculate. As a result, how much shares are bought in each transaction could be ignored.

10.3.1 Simulation Results

Table 10.7 shows the results of the market simulation. From the table, Simulation 1 far outperforms Simulation 2. In order to see whether the earning from the proposed system is statistically significant, another 1000 simulations are conducted. In these simulations, the decision of buying and selling were made at the same time as the proposed system, but without references to the actual content of the news articles, i.e. the decision is completely random. I then compared the distribution of cumulative earnings produced from the randomized trials and Simulation 1. For the randomized system, there are only 38 out of 1000 trials that have a rate of return exceed . Thus, the proposed system is significant at the 0.5% level.

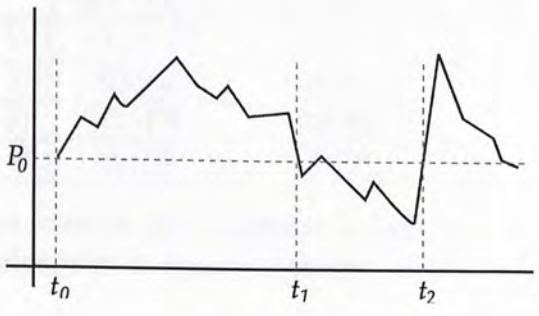


Figure 10.5: A simple diagram illustrates the meaning of hit rate.

10.3.2 Hit Rate Analysis

Hit rate is another important measurement for the predictability of a forecasting system, especially for the proposed system in this paper. It indicates how often the sign of return is correctly predicted. Figure 10.5 illustrates this idea. Assume that at t_0 a prediction which states that the stock price will go upward is made. Since from t_0 to t_1 (T_1), the stock prices are above p_0, we conclude that the prediction is correct in this period, i.e. *hit*. However, from t_1 to t_2 (T_2), the stock prices are below p_0, we therefore conclude that the prediction is wrong in T_2, i.e. *missed*. Thus, if the prediction period, T_m, is varied, different conclusion could be drawn. In other words, the value of m in the market simulation presented in Section 10.3.1 is a critical factor.

Table 10.8 shows the hit rate and rate of return of the proposed system by varying the value of m. The accumulative return and hit rate increase as m increase. It suggested that the system is most stable and suitable for applying the prediction within 3-5 days. It also suggested that such kinds of movements should be tertiary movements, which coincident with the objective described in

	Hit Rate	Acc. Return	S.D. of Return
1 day ($m = 1$)	51.0%	6.58	**1.147**
3 day ($m = 3$)	63.6%	28.06	2.400
5 day ($m = 5$)	**67.4%**	**36.49**	4.135
7 day ($m = 4$)	55.7%	7.22	3.791

Table 10.8: The hit rate of the proposed system by varying holding period. Here, the return is calculated in rate of return.

Chapter 2.

A careful investigation of the prediction task would realize that one of the major reason for making prediction error is that two articles may be very similar in content, but they may have totally different consequence. For instance, two pieces of news articles would both related to SARS, but one is about "WHO issues travel advisory against Hong Kong" and the other one is "WHO lifts travel advisory against Hong Kong". Obviously, they would have completely opposite effect especially toward the traveling industry. However, such kind of small difference can hardly be noticed by any traditional classification algorithm. Notifying this difference is challenging, and I will leave this to the future work.

Part V

The Final Words

Don't gamble; take all your savings and buy some good stock and hold it till it goes up. If it don't go up, don't buy it.

Will Rogers

Chapter 11

Conclusion and Future Work

Scholars and professionals from different areas have already suggested that the relationship between mass media and financial markets is very high. This dissertation further focuses on predicting the impact of the events broadcasted by mass media on market's movements based solely on time series mining and text mining techniques. Here, real-time news articles and intra-day stock prices denote the events broadcasted by mass media and the market's movements, respectively. These data are chosen because they are readily available and the evaluation results obtained can easily be verified.

Several data mining and text mining techniques are in-corporate in the system architecture. The tertiary movements on the stock price movements are identified by a novel piecewise linear approximation approach – the t-test based split and merge segmentation algorithm, which is proposed in Chapter 6.

Chapter 7 describes a heuristics for selecting news articles that are highly related to the tertiary movements of stock trends. News articles are aligned to the stock trends based on the Efficient Market Hypothesis. A χ^2 estimation is

computed for selecting positive training articles. It is computed by considering the features distribution over the entire news collection. The negative training articles are generated based on another selection heuristics that employ the idea of filtering.

According to the contents of the positive and negative training articles, as well as the stock trends, the relationship between news articles and price movements are learned through a new classification approach – Discriminative Category Matching (DCM). Issues related to DCM is presented in Chapter 8 and Chapter 9. DCM does not need to generate any sophisticated model but only requires to calculate simple statistical data. In addition, it does not require any advance document preprocessing, such as feature selection or feature weighting. It consumes very little computational resource for both training and operation. Furthermore, its accuracy is comparable to that of support vectors machine (SVM), the best reported algorithm for text classification up-to-date. For the computational time, DCM depends on the number of categories and the number of features in the collection. Online classification is highly feasible as its response time is short.

Finally, Chapter 10 evaluates various aspect of the proposed prediction framework, such as the quality of time series segmentation, effectiveness and efficiency of DCM. A market simulation using some simple trading strategies is conducted to evaluate the proposed framework. The encouraging results obtained indicated that the proposed framework is highly feasible.

For the future work, this piece of research could be extended in several directions. Currently, the proposed framework is an offline one. Making the model to be able to apply online is an interesting and challenging task. In addition,

investigating the time when the model should be updated is another emerging topic. For the classification model, DCM, its performance could be improved if there exists a heuristics to remove the useless or outdated features incrementally. Lastly, a combination of the traditional time series forecasting techniques, such as moving average and autoregressive, for predicting the market's movements would be another practical research problem.

Appendix

A Hong Kong Stocks Categorization Powered by Reuters

Reuters has categorized all of the Hong Kong stocks into one of the following 12 categories:

Category	Code
Consolidate (A–G)	CONSA
Consolidate (H–O)	CONSH
Consolidate (P–Z)	CONSP
Financial	FINN
Growth Rate Enterprise	GEM
Hotel	HOTL
Industry (A–G)	INDSA
Industry (H–O)	INDSH
Industry (P–Z)	INDSP
Miscellaneous	MISC
Properties	PROP
Utility	UTIL

Table 11.1: The category of Hong Kong stocks.

B Morgan Stanley Capital International (MSCI) Classification

All Reuters corporation news can be accessed by industry. The 20,00 companies reported worldwide have been categorized under the Morgan Stanley Capital International (MSCI) classification. The 38 MSCI industry categories and their news codes are:

Topic	Category	Code
Consumer goods	Appliances and household durables	APL
	Automobiles	AUT
	Beverages and tobacco	BEV
	Food and household products	FOD
	Health and personal care	DRU
	Recreation, other consumer goods	REC
	Textiles and apparel	TEX
Services	Broadcasting and publishing	PUB
	Business and public services	BUS
	Leisure and tourism	LEI
	Merchandising	RET
	Telecommunications	TEL
	Transport – airlines	AIR
	Transport – road and rail	RRL
	Transport – shipping	SHP
	Wholesale and international trade	WHO
Energy	Energy sources	ENR
	Utilities – electrical and gas	ELG
Materials	Building materials and components	BLD
	Chemicals	CHE
	Forest products and paper	TIM
	Metals – non-ferrous	MET
	Metals – steel	STL
	Miscellaneous materials and commodities	MIS

continued on next page...

... continued from previous page

Topic	Category	Code
Capital Equipment	Aerospace and military technology	AER
	Construction and housing	CON
	Data processing and reproduction	DPR
	Electrical and electronics	ELC
	Electronic components/instruments	ELI
	Energy equipment and services	ENQ
	Industrial components	IND
	Machinery and engineering	MAC
Finance	Banking	BNK
	Financial services	FIN
	Insurance	INS
	Real estate	REA
	Multi-industry gold mines	GDM

Table 11.2: Morgan Stanley Capital International (MSCI) classification.

C Precision, Recall and F1 measure

	d belongs to C	d does not belongs to C
d is Assigned to C	a	b
d is not assigned to C	c	d

Table 11.3: A 2×2 contingency table for classification evaluation. Here, d is a testing document and C is a specified category.

Classification effectiveness (quality) is measured in terms of the classic IR notions of precision (ρ) and recall (π). For each category, a 2×2 contingency table as same as Table 11.3 could be formulated.

Refer to table 11.3, ρ and π is calculated from the following equation:

$$\rho = \begin{cases} a/(a+b) & \text{if } a+b>0, \\ 1 & \text{otherwise} \end{cases} \tag{11.1}$$

$$\pi = \begin{cases} a/(a+c) & \text{if } a+c>0, \\ 1 & \text{otherwise} \end{cases} \tag{11.2}$$

F1 measure is the homogeneous averaging over ρ and π:

$$\mathrm{F1} = \frac{2\rho\pi}{\rho+\pi} \tag{11.3}$$

There are two kinds of evaluation approaches using the above methods: micro-level and macro-level. In macro-level, one contingency table per category is used, and the local measures are computed first and then averaged over categories. In micro-level, the contingency tables of individual categories are merged into a single table where each cell a, b, c and d is the sum of the corresponding cells in the local tables.

Bibliography

[1] S. B. Achelis. *Technical Analysis from A to Z.* Irwin, 1995.

[2] P. A. Adler and P. Adler. The market as collective behavior. In P. A. Adler and P. Adler, editors, *The Social Dynamics of Financial Markets*, pages 85–105. Jai Press Inc., 1984.

[3] C. Apte, E. J. Damerau, and S. M. Weiss. Automated learning of decision rules for text categorization. *ACM Transaction on Information System*, 12(3):233–251, 1994.

[4] L. D. Baker and A. K. McCallum. Distributional clustering of words for text categorization. In *Proceedings of the 21th International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, Australia, 1998.

[5] R. J. Bauerm Jr and J. R. Dahlquist. *Technical Market Indicators.* John Wiley and Sons, Inc., 1999.

[6] H. Blumer. Outline of collective behavior. In R. R. Evans, editor, *Readings in Collective Behavior*, pages 22–45. Chicago: Rand McNally College Pub. Co, second edition, 1975.

[7] M. F. Caropreso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In A. G. Chin, editor, *Text Databases and Document Management: Theory and Practice*, pages 78–102. Idea Group Publishing, 2001.

[8] S. Chakrabarti, S. Roy, and M. V. Soundalgekar. Fast and accurate text classification via multiple linear discriminant projections. In *Proceedings of the 28th Very Large Database Conference*, pages 658–669, Hong Kong, China, 2002.

[9] J. M. Clark. Economics and modern psychology. *Journal of Political Economy*, 26:136–166, 1918.

[10] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, 2002.

[11] W. B. Croft. Boolean queries and term dependencies in probabilistic retrieval models. *Journal of the American Society for Information Science*, 37(2):71–77, 1983.

[12] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[13] R. O. Duda and P. E. Harts. *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[14] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, Maryland, USA, 1998.

[15] R. D. Edwards and J. Magee Jr. *Technical Analysis of Stock Trends.* Springfield, fifth edition, 1966.

[16] T. Fawcett and F. J. Provost. Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, pages 53–62, San Diego, California, USA, 1999.

[17] L. Festinger. A theory of social comparison process. *Human Relations*, 7:117–140, 1954.

[18] L. Festinger. *A theory of cognitive dissonance.* Stanford, Calif.: Stanford Univesity Press, Reprinted in 1968.

[19] W. B. Frakes. Stemming algorithm. In W. B. Freakes and R. Baeza-Yates, editors, *Information Retrieval Data Structures & Algorithms*, pages 131–160. Prentice Hall PTR, 1992.

[20] N. Friedman, D. Giger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(203):131–163, 1997.

[21] N. Futr, S. Hartmann, G. Knorz, G. Lustig, M. Schwanter, and K. Tzeras. AIR/X – a rule-based multistage indexing system for large subject fields. In *Proceedings of the 3rd International Conference on Intellignet Text and Image Handling*, pages 606–623, Barcelona, Catalunya, Spain, 1991.

[22] W. R. Greiff. A theory of term weighting based on exploratory data analysis. In *Proceedings of the 21th International Conference on Research and Development in Information Retrieval*, pages 11–19, Melbourne, Australia, 1998.

[23] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning – Data Mining, Inference and Prediction.* Springer, 2001.

[24] J. D. Holt and S. M. Chung. Efficient mining of association rules in text databases. In *Proceedings of 8th International Conference on Information and Knowledge Management*, pages 234–242, Kansas City, Missouri, USA, 1999.

[25] T. Joachims. Making large-scale svm learning practical. Technical Report LS-8 (24), Computer Science Department, University of Dortmund, 1998.

[26] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, Germany, 1998.

[27] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani. An online algorithm for segmenting time series. In *Proceedings of the 1st IEEE International Conference on Data Mining*, pages 289–296, San Jose, California, USA, 2001.

[28] M. Klausner. Sociological theory and the bechavio of financial markets. In P. A. Adler and P. Adler, editors, *The Social Dynamics of Financial Markets*, pages 57–81. Jai Press Inc., 1984.

[29] F. Klein and J. A. Prestbo. *News and the Market.* Chicago: Henry Regenry, 1974.

[30] V. Lavrenko, M. D. Schmill, D. Lawire, P. Ogivie, D. Jensen, and J. Allan. Mining of concurrent text and time series. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining Workshop on Text Mining*, pages 37–44, Boston, MA, USA, 2000.

[31] G. Lebon. *The Crowd: A Study of the Popular Mind.* New York: Macmillan Company, 1896.

[32] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th International Conference on Research and Development in Information Retrieval*, pages 37–50, Copenhagen, Denmark, 1992.

[33] D. D. Lewis. A sequential algorithm for training text classifiers. In *Proceedings of the 17th International Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, Ireland, 1994.

[34] D. D. Lewis. The independence assumnption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, pages 4–15, Chemnitz, Germany, 1998.

[35] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *The 15th National Conference on Artifical Intelligence Workshop on Learning for Text Categorization*, pages 41–48, Madison, Wisconsin, USA, 1998.

[36] D. Meretakis, D. Fragoudis, H. Lu, and S. Likothanassis. Scalable association-based text classification. In *Proceedings of the 10th International Conference on*

Information and Knowledge Management, pages 5–11, Atlanta, Georgia, USA, 2001.

[37] D. Meretakis, H. Lu, and B. Wuthrich. A study on the performance of large bayes classifier. In *Proceedings of the 11th European Conference on Machine Learning*, pages 271–279, Barcelona, Catalonia, Spain, 2000.

[38] D. C. Montogomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., second edition, 1999.

[39] S. A. Nelson. *The ABC of Stock Market Speculation*. Fraser Publishing, third edition, 1903.

[40] H. T. Ng, W. B. Goh, and K. L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th International Conference on Research and Development in Information Retrieval*, pages 67–73, Philadelphia, PA, USA, 1997.

[41] K. Nigam, J. Lafferty, and A. McCallum. Using maximunm entropy for text classification. In *Proceeding of the 16th International Joint Conference Workshop on Machine Learning for Information Filtering*, pages 61–67, Stockholm, Sweden, 1999.

[42] T. Pavlidis and S. L. Horowitz. Segmentation of plane curves. *IEEE Transactions on Computers*, c23(8):860–870, 1974.

[43] D. Permunetilleke and R. K. Wong. Currency exchange rate forecasting from news headlines. In *Proceedings of the 13h Australian Database Conference*, pages 131–139, Melbourne, Australia, 2002.

[44] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MST-TR-98-14, Microsoft Research, 1998.

[45] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21th International Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 1998.

[46] Y. Qu, C. Wang, and X. S. Wang. Supporting fast search in time series for movement patterns in multiples scales. In *Proceedings of the 7th International Conference on Information and Knowledge Management*, pages 251–258, Bethesda, Maryland, USA, 1998.

[47] M. E. Ruiz and P. Srinivasan. Hierarchical neural networks for text categorization. In *Proceedings of the 22th International Conference on Research and Development in Information Retrieval*, pages 281–282, Berkeley, California, USA, 1999.

[48] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[49] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Process Management*, 24(5):513–523, 1998.

[50] H. Schutze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th International Conference on Research and Development in Information Retrieval*, pages 229–237, Seattle, Washington, USA, 1995.

[51] F. Seabastiani. Machine learning in automated text categorization. *ACM Computing Surverys*, 34(1):1–47, 2002.

[52] N. A. Syed, H. Liu, and K. K. Sung. Incremental learning with support vector machines. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, pages 313–321, San Diego, California, USA, 1999.

[53] J. D. Thomas and K. Sycara. Integrating genetic algorithms and text learning for financial prediction. In *Proceedings of the Genetic and Evolutionary Computing 2000 Conference Workshop on Data Mining with Evolutionary Algorithms*, pages 72–75, Las Vegas, Nevada, USA, 2000.

[54] R. H. Turner. Collective behavior. In R. E. L. Faris, editor, *Handbook of Modern Sociology*, pages 382–425. Chicago: Rand McNally & company, 1964.

[55] L. Tvede. *The Psychology of Finance.* John Wiley and Sons, Inc., revised edition, 2002.

[56] K. Tzeras and S. Hartmann. Automatic indexing based on Bayesian inference networks. In *Proceedings of the 16th International Conference on Research and Development in Information Retrieval*, pages 22–34, Pittsburgh, PA, USA, 1993.

[57] C. J. van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119, 1977.

[58] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[59] C. Wang and X. S. Wang. Supporting content-based searches on time series via approximation. In *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*, pages 69–81, Berlin, Germany, 2000.

[60] B. Wuthrich. Probabilistic knowledge bases. *IEEE Transactions of Knowledge and Data Engineering*, 7(5):691–698, 1995.

[61] B. Wuthrich. Probabilistic knowledge bases. *International Journal of Intelligent Systems in Accounting Finance and Management*, 6:269–277, 1997.

[62] B. Wuthrich, D. Permunetilleke, S. Leung, V. Cho, J. Zhang, and W. Lam. Daily prediction of major stock indices from textual www data. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 364–368, New Youk, USA, 1998.

[63] K. Yamamoto, S. Masuyama, and S. Naito. Automatic text classification method with simple class-weighting approach. In *Proceedings of the 3rd Natural Language Processing Pacific Rim Symposium*, pages 498–503, Seoul, Korea, 1995.

[64] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, California, USA, 1999.

[65] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, Nashville, Tennessee, USA, 1997.